SAI-82-738-HU
009/53(241)

REEDS COMPUTER CODE

FINAL REPORT

(NASA-CR-161956) REEDS COMPUTER CODE Final Report (Science Applications, Inc., Huntsville, Ala.) 219 p HC A10/MF A01 CSCI 09B

N82-16760

Unclas
G3/61 07990

23 DECEMBER 1981

NAS8-31806

PREPARED BY:

CHARLES BJORK

Science Applications, Inc.
2109 W Clinton Ave.
Huntsville, Alabama

SAI

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1.0 INTRODUCTION

The REEDS computer code is a tool useful for the estimation of certain rocket exhaust effluent concentrations and dosages and their distributions near the earth's surface following a rocket launch event. Output from REEDS has been useful in producing near-real-time air quality and environmental assessments of the effects of certain potentially harmful effluents, namely HCL, Al_2O_3, CO, and NO. REEDS is the last in an evolutionary series of codes which treats the atmosphere within the first few kilometers of the earth's surface as a single layer; thus the name REEDS - Rocket Exhaust Effluent Diffusion Single layer model.

REEDS was developed for the NASA tropospheric environmental program by the NASA/Marshall Space Flight Center (MSFC) Environmental Effects Task Team (EETT) member, Science Applications, Inc. NASA leadership was provided by the EETT leader, Dr. J. Briscoe Stephens. As a result of continuing research, the function of REEDS is being largely replaced by the more comprehensive multi-layer diffusion program, REEDM. Therefore, this document should valuably assist the process of evaluating the progress made by the application of new techniques to the assessment problem.

REEDS is designed to operate on a Hewlett Packard 1000 series computer system. Research, development and testing took place on the MSFC system at the Space Sciences Laboratory (SSL), followed by implementation and subsequent operational monitoring support and bio-medical studies via the system at the Kennedy Space Center (KSC). A modest effort was made to keep these systems compatible for modeling software operations purposes, even though their diverse purposes obviously gave the systems different capabilities, peripheral hardware, and software.

An important feature of the HP1000 series computers used in the REEDS code is the Program Scheduling feature. The feature permits one self-contained program to initiate another completely self-contained program, and then return operation to the first. This is somewhat analogous to the FORTRAN feature of calling SUBROUTINES. Thus, a program which completely fills the available program memory may execute another, which may execute another, etc., greatly expanding the capabilities of the computer system. Thus, the main executive module schedules major modeling modules, which, in turn schedule other modules, facilitating a logical division of research and modeling effort. The resulting overall structure in the REEDS application is illustrated in Figure 1-1.

REEDS was originally developed for interactive graphical operation through an SAI Plasmascope™*. The requirement for compatibility with the KSC installation necessitated the development of a graphics handler permitting operation with or without the Plasmascope™. This handler uses several assembly-level programmed subroutines, which constitute the only non-FORTRAN software in the package, and which permit complete operation via the HP Cathode Ray Tube (CRT) terminals as well as the Plasmascope™.

While the REEDS code is set up primarily for operation on Titan, Delta, and Space Shuttle Vehicles at their respective KSC/Cape Canaveral Air Force Station (CCAFS) sites, it has a capability for far more general applications. The vehicle data, primarily source rate and altitude histories, are input in parametric form. Simple replacement of these and the site coordinates allows the user to apply the code to a far more general class of problems. This is especially true since the code can handle both the continuous and instantaneous sources.

Further documentation regarding structure and modeling will be found in Section 2 of this document. Operational guidance is contained in Section 3, and program and input data file listings will be found in Section 4.

*Trademark of an SAI Technology Division Product.

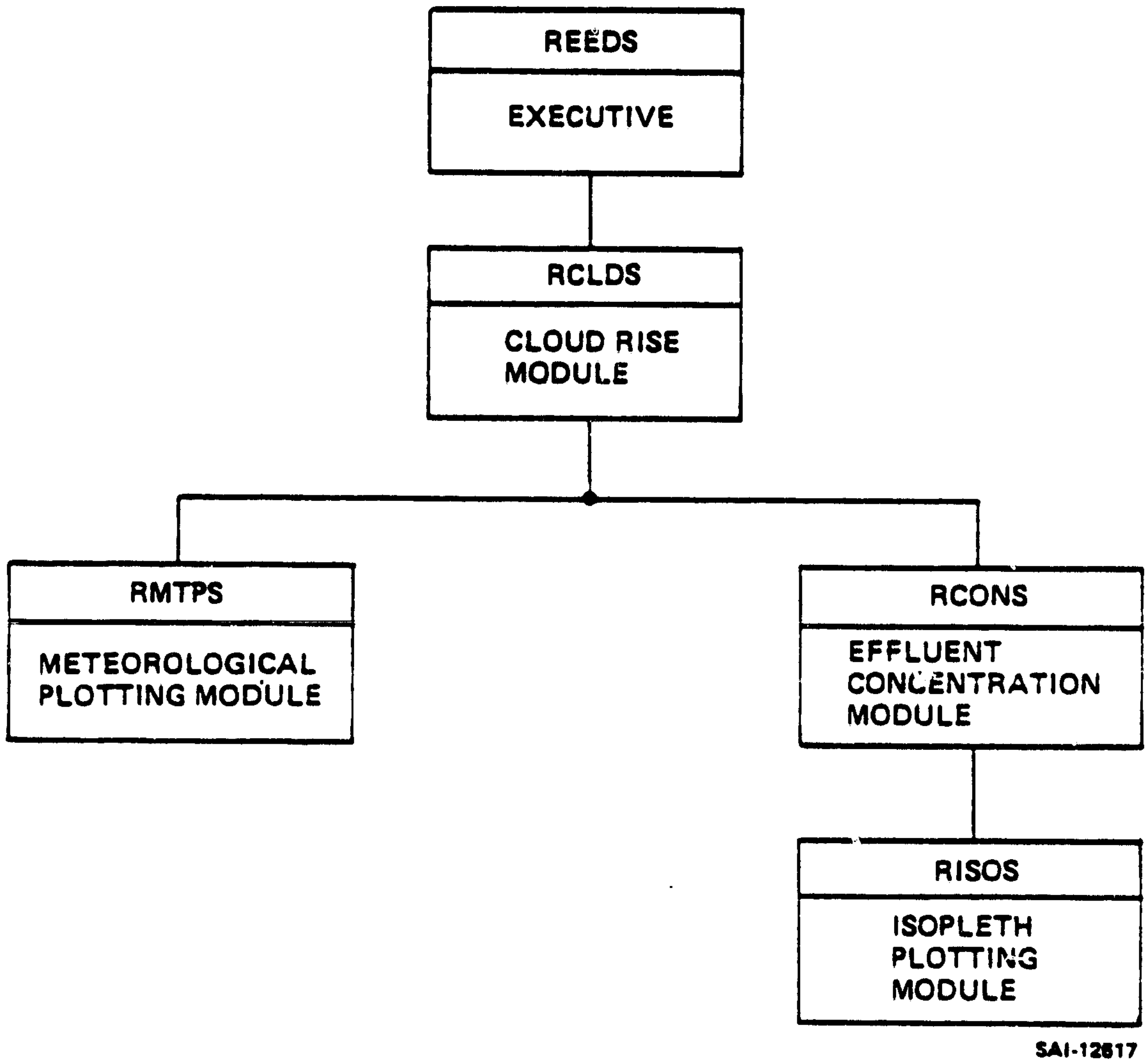


FIGURE 1-1. REEDS PROGRAM SEGMENT SCHEDULING HIERARCHY

SECTION 2.0

REEDS MODEL DESCRIPTION AND BASIC STRUCTURE

The overall structure of the REEDS code is broken down into blocks which largely mirror the phenomenological categorization of the effluent diffusion processes. There is first a "Thermodynamic Phase" in which a rocket exhaust cloud is generated, transported and expanded until approximate thermodynamic equilibrium with its environment is reached (Figure 2-1). The RCLDS module (cloud rise module) contains most of the modeling for this phase. RCLDS calls RMTPS, which illustrates the local meteorology on a plotter for the user, showing approximate geometric relationships with respect to the effective atmospheric boundary layer interface. Then, thermal equilibrium having been reached, the turbulent diffusion process is allowed to proceed (Figure 2-2) in the "Kinetic Phase". The RCONS module contains most of the modeling for this process, and plots the concentration and dosage as a function of range along the cloud centerline. RCONS also calls RISOS, which plots concentration and dosage contours (called "ISOPLETHS") on a map of the launch vicinity. Of course, control, meteorological, and configuration input is provided by the main REEDS executive module, as is sequence control.

To make input more flexible, use is made of "question" input files. For each module for which it is necessary, user input prompts are stored in an indexed file, from which a module may draw in order to prompt the user for a particular input. Changes and/or additions may then be made as necessary to these files. Listings of these files are shown in Section 4.

The structured diagram of Figure 2-3 illustrates the gross operational sequence of the entire code. This will be elaborated upon in the following subsections.

2.1 MODULE DESCRIPTION

Each module will now be roughly described in narrative fashion, seasoned with the appropriate mathematical expressions employed in the modules. As much as possible, the narrative "walk" through each module will follow the commenting in the code. Some commenting not directly related to calculational procedures may be excluded from the description. In addition to making use of

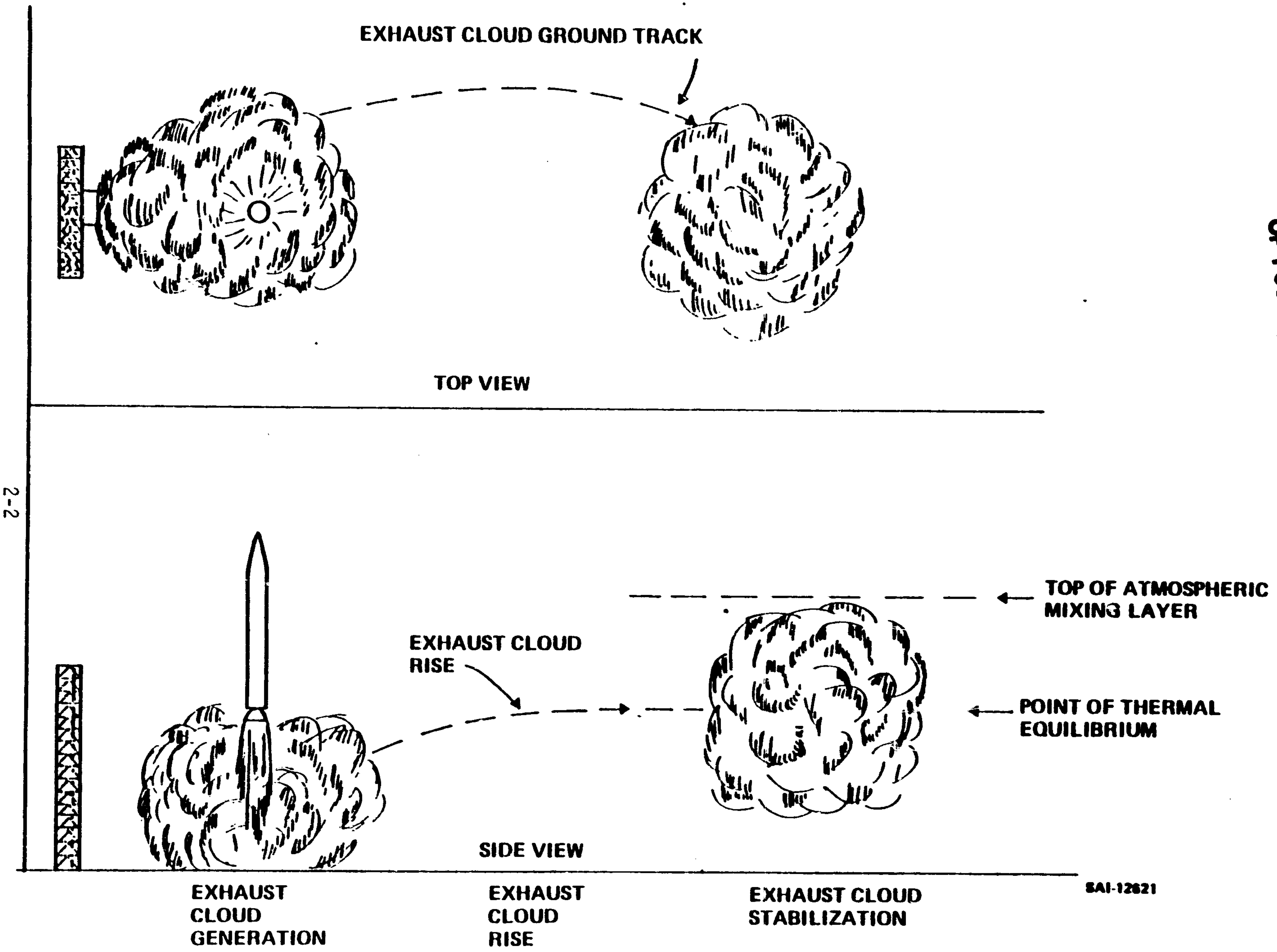


FIGURE 2-1. ILLUSTRATION OF THE "THERMODYNAMIC" PHASE MODELED IN REEDS

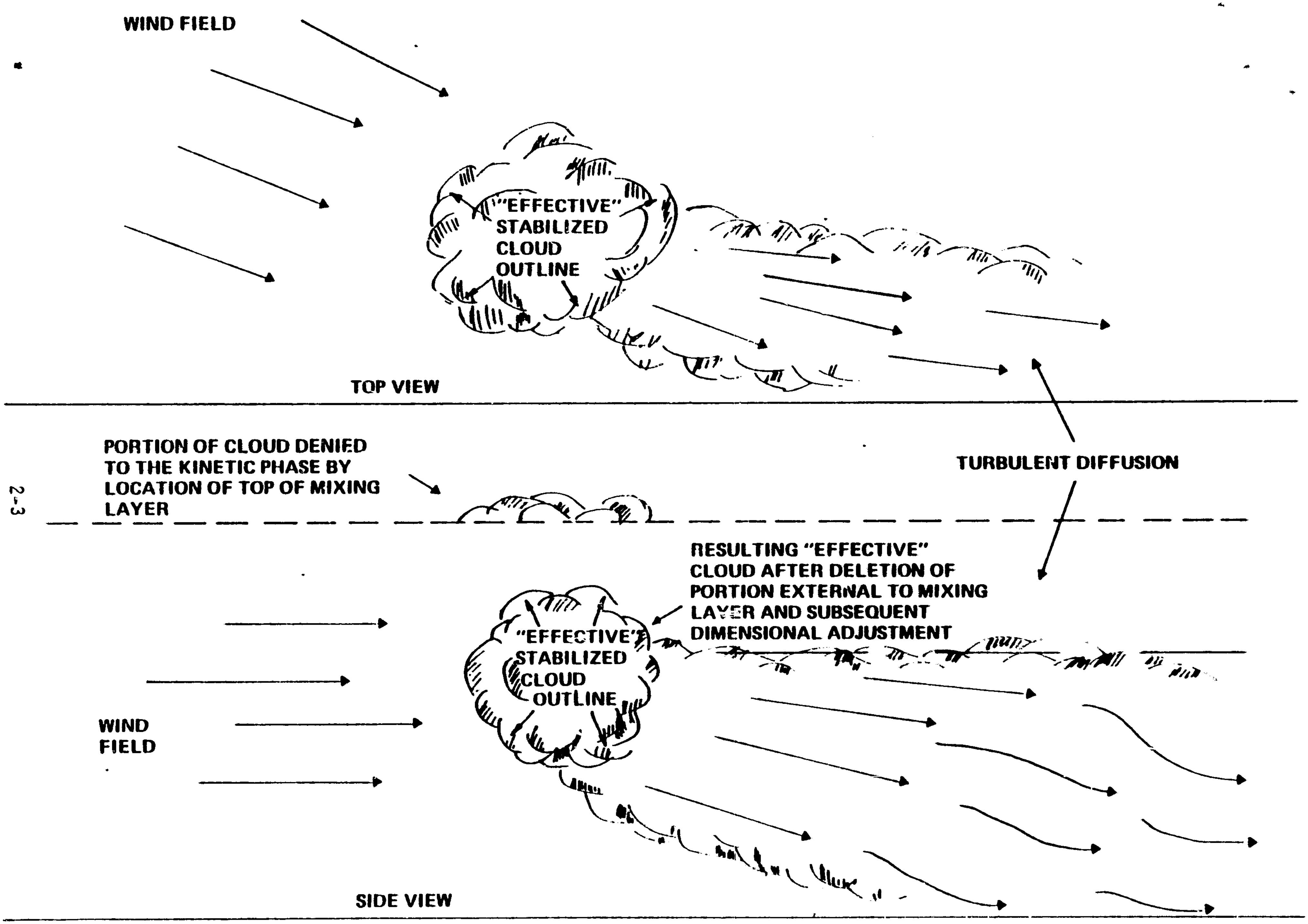


FIGURE 2-2. ILLUSTRATION OF THE "KINETIC" PHASE MODELED IN REEDS

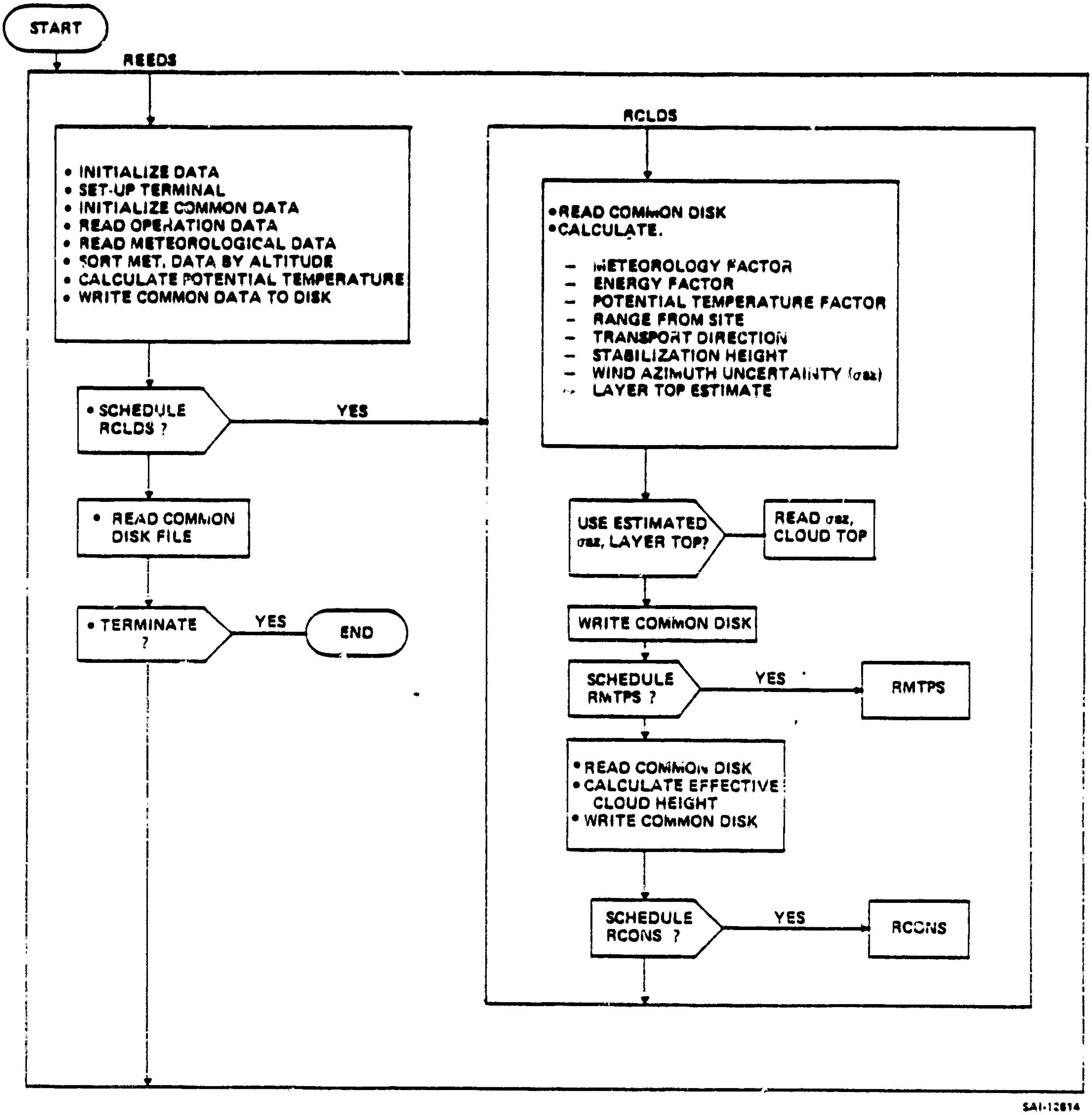


FIGURE 2-3. STRUCTURED DIAGRAM OF THE REEDS CODE SHOWING IMPORTANT AND COMMENTED OPERATIONS

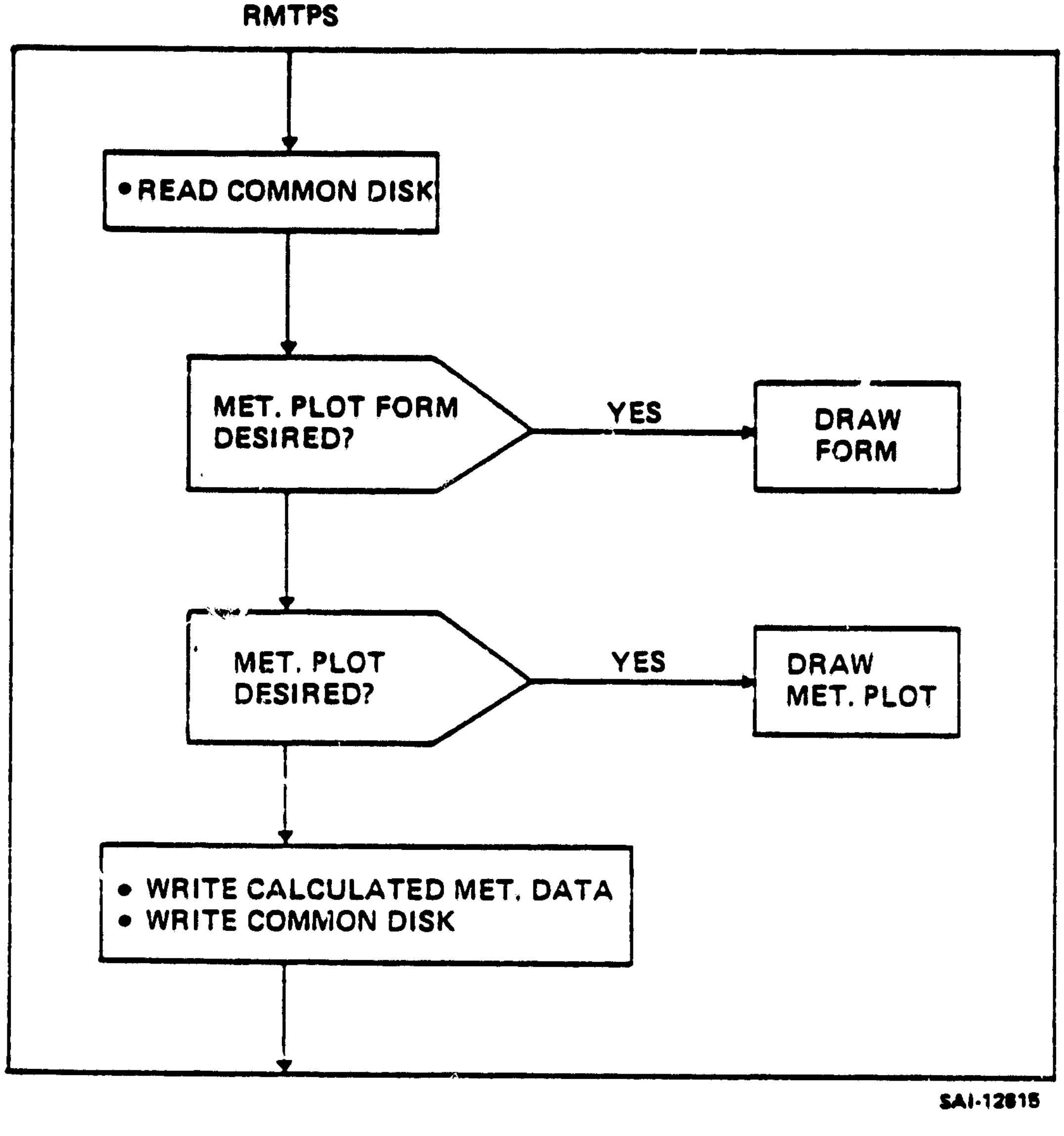


FIGURE 2-3. (Continued) STRUCTURED DIAGRAM OF REEDS CODE

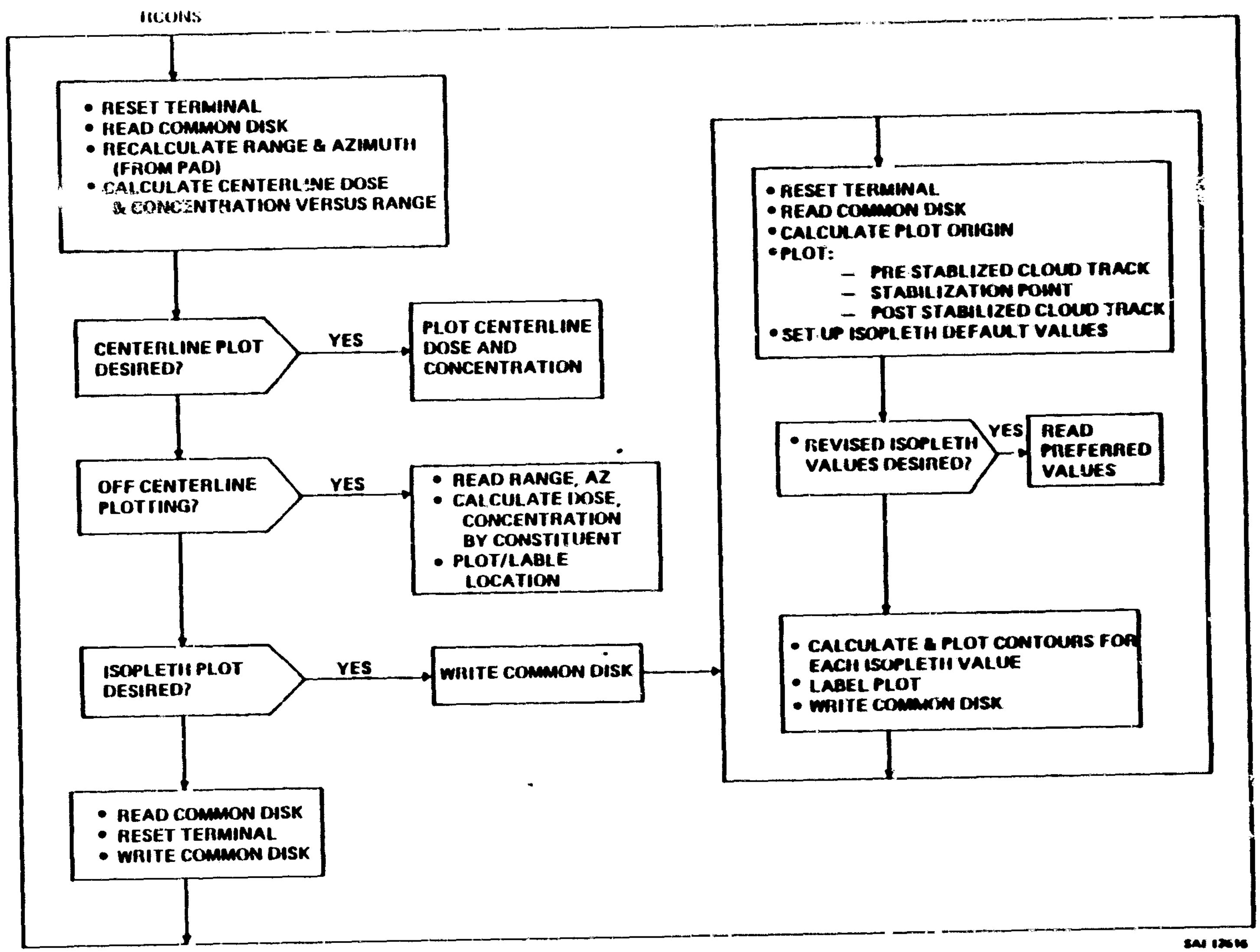


FIGURE 2.3. (Concluded) STRUCTURED DIAGRAM OF REEDS CODE

the commenting in the code as "landmarks", use will be made of the structured diagram of Figure 2-3 for guidance in module description. Each major calculational and structural event or user input option will be noted as a paragraph heading throughout these descriptions.

Common Data. The REEDS code makes use of bulk or "blank" common to simplify the use of subroutines. A penalty was paid for this in the form of a slight loss of modularity and some pain and agony due to vulnerability to "side effects" (indirectly and inadvertently changing a variable during a modification/upgrade). The penalty was lessened by structuring the common into testable "pseudo" blocks (not to be confused with block or labeled common), and by the use of descriptive comments for each significant "pseudo" block. This data is shared by the module and some of its subroutines, and is written to a "common" disk file. The common disk file is created by the program with the name "=REEDS" when the first attempt is made to write the common data. It is finally destroyed (purged) by the program upon successful termination of the program. If an abnormal termination leaves the file unpurged, it is, in principle, written over when it is detected to pre-exist; however, some strange results have occasionally been produced when the file pre-existed, prompting prudent users to be sure that it was purged before execution. Thus, following a brief statement of purpose, a date for last modification and any necessary loading instructions, a large block of common data (316 words) appears in each module. Just before another module is called this data is written on the disk file. Then, as soon as the new module is in core, the disk file is read, reloading the common. When the new module concludes, or prepares to call still another module, the common data is written again. A brief description of the common variables (Figure 2-4) appears in Table 2-1.

The REEDS "Question" Files. Several REEDS modules have their own question files, within which are stored informative messages and/or user prompts. These are indexed sequential coded (and therefore listable) files, program access to which is accomplished by calling subroutine "DREAD" (for Disk READ). The arguments to DREAD are:

1) A six (6) character, three-word file name

```
C
C  >>>   COMMON AREAS
C
C
C  >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C  >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNOFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,MONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C  >>>   CALCULATION TIME
      COMMON ITIME,IDAY,INONTH(2),IYEAR
C
C  >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C  >>>   LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME, LVERSN
C
C  >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C  >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C  >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
```

Figure 2-4. Program Excerpt Showing the Common Data Arranged in "Pseudo" Blocks

TABLE 2-1 COMMON DATA DESCRIPTION

"PSEUDO" COMMON BLOCK	PROGRAM LABEL/ EQUIVALENCE NAME	SYMBOL	DESCRIPTION
MATH PARAMETERS	PI	π	3.141593
	PIOVR2	$\pi/2$	.5 • π
	PI43	$\frac{4}{3}\pi$	1.33333 • π
	TWOPI	2π	2.0 • π
	SQR2PI	$\sqrt{2\pi}$	$\sqrt{\text{TWOPI}}$
	DEGRAD	$\pi/180$	DEGREES-TO—RADIANS CONVERSION PARAMETER
	RADDEG	$180/\pi$	RADIANS-TO-DEGREES CONVERSION PARAMETER
	IV2	—	CONVENIENT INDEX
VEHICLE PARAMETERS	VPAR(1)/QC1	$\dot{Q}_{C_1}$	TOTAL SOURCE OUTPUT RATE (g/sec), NORMAL BURN
	VPAR(2)/QC2	$\dot{Q}_{C_2}$	TOTAL SOURCE OUTPUT RATE (g/sec), ABNORMAL-SINGLE ENGINE BURN
	VPAR(3)/QC3	$\dot{Q}_{C_3}$	TOTAL SOURCE OUTPUT RATE (g/sec), ABNORMAL-EXPLOSION/BURN
	VPAR(4)/QT1	Q_{C_1}	TOTAL SOURCE STRENGTH (g), NORMAL
	VPAR(5)/QT2	Q_{C_2}	TOTAL SOURCE STRENGTH (g), ABNORMAL-SINGLE ENGINE
	VPAR(6)/QT3	Q_{C_3}	TOTAL SOURCE STRENGTH (g), ABNORMAL-EXPLOSION
	VPAR(7)/AA	a	*1st ROCKET RISE POWER LAW COEFFICIENT
	VPAR(8)/BB	b	*2nd ROCKET RISE POWER LAW COEFFICIENT
	VPAR(9)/CC	c	*3rd ROCKET RISE POWER LAW COEFFICIENT
	VPAR(10)/HEATN	Q_{I_1}	EFFECT HEAT RELEASE (cal/g), NORMAL BURN
	VPAR(11)/HEATM	Q_{I_2}	EFFECTIVE HEAT RELEASE (cal/g), ABNORMAL-SINGLE ENGINE BURN
	VPAR(12)/HEATA	Q_{I_3}	EFFECTIVE HEAT RELEASE (cal/g), ABNORMAL-EXPLOSION
	VPAR(13)/PHCL	f_{HCL}	PARTIAL FRACTION OF HCL
	VPAR(14)/PCO	f_{CO}	PARTIAL FRACTION OF CO
	VPAR(15)/PCO2	f_{CO_2}	PARTIAL FRACTION OF CO_2
	VPAR(16)/PAL203	$f_{Al_2O_3}$	PARTIAL FRACTION OF Al_2O_3
	VPAR(17)/PNO	f_{NO}	PARTIAL FRACTION OF NO
	VPAR(18)/GAMMAX	γ	CLOUD RISE ENTRAINMENT COEFFICIENT
	SIGHCL	σHCL	CONCENTRATION STANDARD DEVIATION OF HCL

SAI-13818

TABLE 2–1. Continued

PSEUDO COMMON BLOCK	PROGRAM LEVEL EQUIVALENCE NAME	SYMBOL	DESCRIPTION
OPTION PARAMETERS	YPOS	—	CURRENT TERMINAL DISPLAY CURSOR ORDINATE
	YES	—	LOGICAL "YES" CODE (= 1 HY)
	ISNDFO	—	METEOROLOGICAL DATA TYPE INDICATOR 0 = SOUNDING 1 = FORECAST
	IVHICL	—	ROCKET VEHICLE INDEX (1, 2, 3, 4, 5 FOR SPACE SHUTTLE, TITAN, DELTA, DELTA 3914, AND MINUTEMAN, RESPECTIVELY)
	IRUN	—	PROGRAM EXECUTION MODE (OPERATIONAL PRODUCTION)
	ICALC	—	CALCULATION FLAG
	IPLACE	—	METEOROLOGICAL DATA LOCATION CODE
	ITIMEZ	—	METEOROLOGICAL DATA TIME ZONE CODE
	IGOTMP	—	FLAG TO INDICATE WHETHER OR NOT A MAP ORIGIN HAS BEEN SET
	NUMRUN	—	TOTAL NUMBER OF RUNS (MET DATA FILE TO PROCESS)
	LU	—	INTERACTIVE TERMINAL DEVICE CODE
	NUM	—	RUN SEQUENCE NUMBER
	MONLY	—	METEOROLOGICAL PLOT SWITCH 1 = YES 0 = NO USED IF ONLY PURPOSE OF RUN IS MET PLOTS
	MFORM	—	METEOROLOGICAL PLOT FORM SWITCH 1 = YES 0 = NO
	LUPRNT	—	PRINTING LOGICAL DEVICE KEY 6 = PRINTER OUTPUT, ELSE NO PRINTER OUTPUT
	ITDU	—	TOWER DATA SWITCH 0 = NO TOWER DATA 1 = TOWER DATA
	IPL1	—	METEOROLOGICAL PLOTTING DEVICE CODE
	IPL2	—	CENTERLINE PLOTTING DEVICE CODE
	IPL3	—	ISOPLETH PLOTTING DEVICE CODE

SAI-12757

TABLE 2–1. Continued

PSEUDO COMMON BLOCK	PROGRAM LEVEL EQUIVALENCE NAME	SYMBOL	DESCRIPTION
LAUNCH TIME	LTIME	—	LAUNCH TIME FLAG (TRUE IF LAUNCH TIME READ; OTHERWISE, FALSE)
	LTIM	—	LAUNCH TIME OF DAY (0001; 2400)
	LDAY	—	LAUNCH DAY OF MONTH (01; 31)
	LMONTH	—	2 WORD ARRAY FOR LAUNCH MONTH OF YEAR (e.g., 2HJA, 1HN)
	LYEAR	—	LAUNCH YEAR (e.g., 1981)
	LAUNTD	—	ARRAY OF CHARACTERS COMPRISING THE LAUNCH DATE AS READ FROM THE TERMINAL
CLOUD STABILIZATION AND LAYER PARAMETERS	STBALT	H	STABILIZATION HEIGHT (METERS)
	STBHT	H_{S_E}	EFFECTIVE STABILIZATION HEIGHT (METERS)
	STBAZD	θ_s	AZIMUTH FROM PAD OF STABILIZATION (DEGREES)
	STBRNG	R_s	RANGE FROM PAD OF STABILIZATION (METERS)
	STBTIM	t_s	TIME FROM LAUNCH OF STABILIZATION (SECONDS)
	CLDRAD	R_c	RADIUS OF CLOUD (METERS)
	RISTIM	Δt_i	30-WORD ARRAY OF TIMES FOR PASSAGE OF CLOUDS THROUGH EACH MET. LAYER
	BOTLAY	Z_B	HEIGHT OF BOTTOM OF LAYER CONTAINING STABILIZATION ALTITUDE
	TPLAY	Z_T	TOP OF LAYER CONTAINING STABILIZATION
	CALHT	Z	CALCULATION HEIGHT (Z_B X $Z \leq Z_T$)
	LAYTOP	—	INDEX OF LAYER TOP ARRAY ELEMENT CONTAINING STABLIZATION
	LAYBOT	—	INDEX OF LAYER BOTTOM ARRAY ELEMENT CONTAINING STABILIZATION
	REFLEC	γ	SURFACE REFLECTION COEFFICIENT 0 = TOTAL REFLECTION 1 = TOTAL ABSORPTION
	IPTZ	—	LEAST SQUARE FIT FLAG FOR POTENTIAL TEMPERATURE GRADIENT CALCULATION 1 = NO FIT 0 = USE THE FIT

SAI-12760

TABLE 2-1. Continued

PSEUDO COMMON BLOCK	PROGRAM LEVEL EQUIVALENCE NAME	SYMBOL	DESCRIPTION
MET DATA	ALT	Z	ARRAY OF LAYER-TOP ALTITUDES IN THE MET PROFILE
	IDIR	θ	ARRAY OF WIND DIRECTIONS OF THE LAYERS OF THE MET PROFILE
	SPEED	u	ARRAY OF WIND SPEEDS OF THE LAYERS OF THE MET PROFILE
	TEMP	T	ARRAY OF TEMPERATURES OF THE LAYERS OF THE MET PROFILE
	PRESS	P	ARRAY OF PRESSURES OF THE LAYERS OF THE MET PROFILE
	PTEMP	PT	ARRAY OF POTENTIAL TEMPERATURES OF THE LAYERS OF THE MET PROFILE
	SURDEN	α_o	SURFACE DENSITY OF THE MET PROFILE
	FILE	—	FILE NAME ARRAY (6 CHARACTERS, 3 WORDS) OF THE MET FILE
CALCULATION TIME	ITIME	—	CALCULATION TIME OF DAY (0001; 2400)
	IDAY	—	CALCULATION DAY OF MONTH (01:31)
	IMONTH	—	2-WORD ARRAY FOR CALCULATION MONTH OF YEAR (e.g., 1981)
SOUNDING TIME	ISTIME	—	SOUNDING TIME OF DAY
	ISDAY	—	SOUNDING DAY OF MONTH (01:31)
	ISMONTH	—	2-WORD ARRAY FOR SOUNDING MONTH OF YEAR (e.g., 2HFE, 1 HB)
	ISYEAR	—	SOUNDING YEAR (e.g., 1981)

SAI-12799

TABLE 2-1. Continued

PSEUDO COMMON BLOCK	PROGRAM LEVEL EQUIVALENCE NAME	SYMBOL	DESCRIPTION
DIFFUSION COEFFICIENTS	SIGXO	σX_{o}	STABILIZED CLOUD ALONG WIND STANDARD DEVIATION
	SIGX	σX	ALONG-AXIS 1-CLOUD EXTENT
	SIGAP	σ_{A}'	WIND AZIMUTH STANDARD DEVIATION AT STABILIZATION
	XDIST	X	ALONG-AXIS DISTANCE
	YDIST	Y	OFF-AXIS DISTANCE
	EXPZ	Σ_{Z}	VERTICAL DOSAGE TERM
	SQSIGZ	$2\sigma Z^{2}$	TWICE THE SQUARE OF THE 1- VERTICAL CLOUD EXTENT
	SIGY	σ_{Y}	OFF-AXIS 1-CLOUD EXTENT
	SIGAL	σ_{AL}	LAYERED—AVERAGED WIND AZIMUTH STANDARD DEVIATION
	CLDLNG	C_{L}	CLOUD LENGTH (METERS)
	AVSPD	U	SPEED OF CLOUD AVERAGED OVER MET. LAYERS (METERS/SEC)
	AVDIR	θ	DIRECTION OF CLOUD AVERAGED OVER MET LAYERS (DEGREES)
	SIGAZ	σ_{AZ}	WIND AZIMUTH UNCERTAINTY (DEGREES)
CONCENTRATION AND DOSAGE VALUES AND RANGES	CONMAX	χ_{M}	MAXIMUM PEAK CONCENTRATION (PPM)
	CONCPK	χ_{p}	PEAK CONCENTRATION (PPM)
	DOSMAX	D_{M}	MAXIMUM PEAK DOSAGE (PPM—SEC)
	DOSPK	D_{p}	PEAK DOSAGE (PPM—SEC)
	RNGSMC	R_{Mc}	RANGE OF MAXIMUM CONCENTRATION (FROM PAD)
	RNGSMD	R_{MD}	RANGE OF MAXIMUM DOSAGE (FROM PAD)

SAI-12780

2) The desired line number (index + 1)

3) A sixty-four (64) character (32 word) data buffer to receive the ASCII data.

The first line of each question file is an eighty (80) character sequence of the numbers

12345678901234567890I2. . .

useful for preparation/editing of the question file itself (i.e., aids in columniation of characters), so the actual line number is always one (1) greater than the index of useful data to be retrieved from the question file. Section 4.3 shows the REEDS question file listing. Special CRT display enhancement characters are often embedded as part of the character strings. They are unprintable with the selected print option used in generating the listings of Section 4, and do not show up even as blank spaces. This gives an erroneous column indication for characters which do show up. However, the special Character Display feature of all HP CRT's permits display of the enhancement character symbols, facilitating editing of the lines of these files. Once read by DREAD, these lines are printed on the desired device by subroutine "CHAR", which performs the necessary terminal-dependent oerations required to display the line read.

Communication of Data To (Or From) Output Files. When it is desirable to communicate program numeric data to (encoding) an output file, or from (decoding) that file, it must be first internally converted to (or from) an ASCII character string. Encoding is accomplished with a call to the system subroutine "CODE" and a subsequent specialized "READ" operation. Decoding requires the use of "WRITE", instead of "READ". The optional argument to CODE is the number of characters desired to be encoded (or decoded). Many painful agonies were avoided by explicitly defining and using this argument. Also great debugging horrors were generated by inadvertently putting some other statement between the call to "CODE" and its subsequent "READ" (or "WRITE"), or otherwise failing to resolve a call to "CODE" with the necessary "READ" (or "WRITE") call. (These are mentioned here in the event users use this document as a reference in making any changes to REEDS.) The arguments to "READ" (or "WRITE") are normally the standard output device variable and the FORMAT code. However, when "READ" (or "WRITE") is used to resolve a call to "CODE" to encode (non-ASCII) or decode (ASCII) data, the first argument is the array variable name used to receive the encoded (or send the decoded) data. The size of this array must be compatible with the FORMAT statement (the 2nd argument) used to accomplish the "READ" (or "WRITE"). Once encoded, the ASCII string is written to an output, whenever it is desired, in the normal fashion, i.e., with an "A" FORMAT formatted "WRITE" call. If the data were decoded (via the "WRITE" call) instead, the resulting data (FORMAT permitting) could be arithmetically or otherwise processed, and must be re-encoded if the ASCII string is needed.

2.1.1 The REEDS Executive Module

The REEDS executive module

- Initializes some global operational, terminal control, and calculational data
- Reads additional control and parameteric data
- Reads and prepares the meteorological data
- Calculates the potential temperature profile
- Controls subsequent scheduling of modules.

The bulk of the module consists of interactive option selection and related parameter initialization. The main program (file "REEDS") is 1157 source lines long at this printing, and the subroutine packages required (files "REEDT", "LINE", "DLINE", "CALIX" and "CTLB2") constitute another 1555 source code lines. Refer to Figure 2-3 as necessary to follow the program structure.

User Terminal Identification Setup. The type of terminal (CRT, Plasmascope) is first discerned by the program from system tables internal to the HP1000 operating system. Some terminal parameters are subsequently initialized -- automatically and transparent to the user. The header is transferred from the REEDS executive module question file (called "*REEDS") to the terminal display.

Cloud Rise Fit or Virtual Potential Temperature Lapse Rate Fit Option. The desired fit to the Lapse Rate of Virtual Potential Temperature is then produced for the cloud rise model as a result of a user option. The choice is between a deterministic method or a least squares method suggested by

EETT leader Dr. J. B. Stephens. The latter method produces consistently lower altitudes than the former for cloud rise stabilization and has thus been termed the "low altitude fit". This choice remains as a user option pending documented validation of one method or the other. This option was installed after Plasmascope operation was largely phased out. Thus, the normal terminal independent input sequence is not used, rather, a direct CRT terminal input is used.

Number of Runs and Input Meteorological Data File Names. The REEDS program was designed to process multiple meteorological data files if that is desired. The procedure is to set up a series of data files which have in common the first four (4) characters of their names, and a relative sequence number, i.e., 01, 02, . . . 99, for their last 2 characters. When prompted to do so, the user enters:

XX, FFFFYY

for the number of runs, data file name root, and code YY, respectively, where

XX	=	Number of runs desired [01 ; 99]
FFFF	=	The first four (4) characters of the series of meteorological data file names
YY	=	The file of the series it is desired to start with, e.g., DATA01

The program has a default data file name root of "DATA" which may be invoked by either assenting to the use of the default (which also assumes there will be only 1 run of DATA01), or by entering blanks for FFFF (which causes any entry for YY to be ignored, presetting its value to 1, thus causing the series DATA01 DATAXX to be executed). One run of an arbitrary file name is accomplished by declining the default, and entering

XX, FFFFYY

where XX = 01, and FFFFYY is the six (6) character file name desired, e.g.,

01, TESTFZ

<u>Geographical Location and Time Zone Codes</u>. As a result of a standard NASA naming convention the geographical location, and therefore the time zone code, can be embedded within the meteorological data file name. These are the common variable ITIMEZ and IPLACE, respectively. The following illustrates the implications of certain non-standard values for FFFF:

FFFF = VAND (Vandenberg AFB), IPLACE = 1, ITIMEZ = 1HP

FFFF = KSC (1965 data tape), IPLACE = 2, ITIMEZ = 1HE

FFFF = KSC, IPLACE = 3, ITIMEZ = 1HE

FFFF = SPEC, Special location, IPLACE and ITIMEZ = 0, and actual subsequently input according to the format: values

:XX,YYZ,AAAA

where

XX = Day of month [01 ; 31]
YYY = Month [Jan ; Dec], and
AAAA = Time [0001 ; 2400].

Any other input is assumed to be the standard KSC meteorological data file with ITIMEZ = 1HE, and IPLACE = 2. The meteorological data file can only be an actual sounding for FFFF = SPEC, whereas for other inputs sounding and/or TOWER data may be used.

<u>Research, Operational, or Production Modes</u>. As of the time this document was being generated the only validated mode of these alternatives was the operational mode -- which is also the default and most commonly used mode. While the additional user inputs inherent to the research mode will be mentioned, their demonstration in Section 3 will have to await a subsequent revision of this document. Nevertheless, the appropriate non-default selection is made by entering a single character R, O, or P, respectively. This results in the

setting of common variable IRUN to 1, 2, or 3, respectively. Default selection is, of course IRUN = 2.

Atmospheric Mixing Layer Top and Wind Azimuth Uncertainty - For Production Runs. For production runs, the input of the top of the atmospheric mixing layer, TOPLAY, (in meters) and the standard deviation of the wind azimuth uncertainty, SIGAZ, (in degrees) must be input from the keyboard. The format, upon prompt is

XXXX.XX

for TOPLAY, with possible range [0;999999.0] meters, followed by

XX

for SIGAZ, with possible range [0;99] degrees. For the other operational modes (Research, Operational) the values of TOPLAY and SIGAZ are estimated and then optionally changed by the user. This will be elaborated upon when it is encountered in the RCLDS cloud rise module (Section 2.1.2)

Meteorological Forms Switch. The REEDs program plots some of the meteorological data (called a "Met Plot") and a schematic of the stabilized exhaust cloud. The form for this plot can also be prepared by the program on the plotter, but the process is very time consuming. Therefore, it is better done as infrequently as is necessary and high-speed reproduction devices used to generate a supply of the forms. For the purpose of generating this form, the REEDS program may be run to the point of this input for the sole purpose of generating the form. The run may then be terminated and subsequently restarted once a supply of forms is available. Thus, when prompted for "Met Plot Forms?", the user may respond "Y" for 'Yes' or "N" for 'No', the latter of which is the default response. This sets the common (OPTION pseudo block) parameter MFORM to 1 or 0, respectively.

Meteorological Plots. The meteorological plot may also be skipped if it is desirable. When prompted, a "Y" or "N" indicates 'Yes' or 'No', respectively. This sets the common OPTION pseudo-block) parameter MONLY to 1 or 0, respectively. The first setting is the default setting.

Line Printer Output. The line printer output may be omitted if desired. When prompted a "Y" or "N" indicates 'Yes' or 'No', respectively. This sets the common (OPTION pseudo-block) parameter to 6 or 64_{10} respectively. The first setting is the default setting, and its value of 6 indicates the standard printing device code. The value of 64_{10} for the alternate setting formerly cleared the parameter (largest value was $77_8 = 63_{10}$). Now, however, the device code range has been extended above 100_8, making 64_{10} a legal device code. However, no such code is active on the MSFC or KSC machines, so no change (e.g., $64_{10} \rightarrow 128_{10}$) has yet been made.

Launch Time and Date. The current computer system time and date are loaded into the launch time and date variables, LTIM, SDT, LDAY, LMONTH, LYEAR via the call to the subroutine GETDT, which reads the system clock. The user is prompted for this information, and if the response to the prompt is "N" for 'No', the system time/date is rejected as the launch time, and opportunity is given for the above variables to be read accordingly to

TTTbbbbbDDbMMMbYYYY

where	TTTT	= the time of day [0001;2400] ,
	b or bbbbb	= blank space place holders indicating the number of user-input blank spaces
	DD	= the day of the month [01;31]
	MMM	= the month [JAN;DEC]
	YYYY	= the year (e.g., 1980)

If the response to the original prompt is "Y" for Yes or blank, the system time is accepted for the default launch time/date, and the time/date variables are left as initialized.

Launch Vehicle. The user is then prompted for the launch vehicle desired. The options are "D" for Delta, "T" for Titan, and "S" for Shuttle. This sets the common (OPTION pseudo-block) variable IVHICL to 0, 1, or 2, respectively.

Meteorological Data Input. If the IPLACE parameter, set by the meteorological data file name read above, is 2, then the forecast/sounding data is assumed to reside on tape and is read from device code 8 via subroutine KSC65 (explained below). Otherwise the appropriate disk file name is opened. Whether the data represents an actual sounding or a forecast is indicated by a code read from the file header. If a sounding is indicated, the data type parameter, ISNDFO, (an OPTION pseudo-block common variable) is set to 0; if it is a forecast, it is set to 1. As the header and/or data lines are read, they are printed on the printer device. The following indicates the sequence or data lines read.

(1) test code, data type and location header information

(2) time of data, ascent and data header information

(3) actual data for each altitude:

altitude (feet)

wind direction (degrees from north)

wind speed (knots),

temperature (^oC),

Dew Point (^oC),

pressure (millibars)

relative humidity (percent)

Absolute humidity (gms/M^3) (if present on file)

Air density (gms/M^3) (if present on file)

other data (if present) (ignored by program)

(4) Terminator (NNNN) indicating end of data. (See Section 4.4)

The program is designed to ignore all non-essential data words and/or data lines. The altitude and wind velocity are then converted to other units; they are converted as follows:

altitude $\rightarrow$ m

wind speed $\rightarrow$ m/sec

All data lines with altitudes less than 10000 feet are then sorted by ascending order of altitude. The potential temperature is then calculated according to

$$PTEMP_i = (TEMP_i + 273.6)\left(\frac{1000}{PRESS_i}\right)^{.288}$$

where PTEMP, TEMP, and PRESS are the COMMON variables (MET data pseudo-block) potential temperature, temperature, and pressure, respectively. The sorted data is then written on the output device and the meteorological data file closed.

Met File Reads Via Subroutine KSC65. The subroutine KSC65 handles MET file reads from tape (device code 8). The sequence of data is identical to that of the disk file read. The subroutine was designed to read large data bases, especially the 1965 KSC meteorological data tapes (thus the subroutine name). The arguments to KSC65 facilitate selection of specific data sets from the tape. They are as follows:

IBUF = Eighty (80) character (40 word) integer buffer array to hold ASCII data lines read from device 8 pending decoding.

IALT = Altitude data array to be passed back to the main program for eventual conversion to the analogous common array.

DPTEMP = dew point temperature array to be passed back to the main program

IFOFF1 = (KSC65 variable IWANT) = ordinal data set indicator for selection among data sets of the same date/time.

IEOF = End of File (EOF) flag set if EOF encountered on device 8 (set by Equipment Table Check Three system EXEC call)

ITIMG = Sounding time desired

IDAYG = Sounding day desired

IMONG = Sounding month desired

ISNDG = data type desired (sounding/forecast)

RUNNUM = run numbers desired

Creation of the Common Data Disk File. The common data, though not completely defined yet, is written to the disk file "=REEDS" via a call to the subroutine "RWDIS." The two arguments to RWDIS are the six (6) character common data disk file name (3 integer words, 2 characters each), and the operation integer code (1 = write).

RWDIS contains the 316 word common block as it exists in the main program. In addition, space is provided for the data control block array (locally called ODCB) necessary to control the disk read (transparent to user - only provision of the space necessary, i.e., 144 integer words or more for the control array). The data is transferred to/from disk via a buffer array. While the common data transferred is 316 words, disk storage is only by integer word equivalents. Thus, because of the real data variables/arrays, the integer word equivalent of the 316 words is 557 words. The buffer array is thus 557 words. The ISIZE array, which is the parameter controlling the file creation operation, must have its two integer words set to 30 and 557, respectively. The first word is the number of blocks the 557 words are written in.

The file is first opened. If it is not detected to pre-exist, the file is created with the CREAT system call. For a RWDIS operation code of 1, the file is written; if 0, the file is read. The file is then closed.

Transfer to the Cloud Rise Module. As can be seen in Figure 2-3, transfer to the Cloud Rise Module RCLDS then occurs. The transfer is accomplished via the system EXEC call, which schedules the program whose name is contained in a six (6) character (3 word) argument array (RCLDR). The program must have an ID segment in CORE, an operation accomplished via the File Manager RP command which must be executed before the running sequence begins (see File Manager or Terminal Users manual).

Program Termination. At the conclusion of RCLDS, transfer is returned to REEDS and the user is prompted to decide for program termination ("Y" or default response) or for the processing of more data ("N" for No termination). If the latter response is chosen, REEDS begins again. Otherwise, the CRT screen is cleared, and the common disk file is purged. Finally, all program ID segments in the REEDS series are removed from core ("OFF'ed" -- the reverse of the "RP" operation -- see File Manager or Terminal User's manual). The program STOP is executed and REEDS is terminated.

2.1.2 The RCLDS Cloud Rise Module

The RCLDS Cloud Rise Module models the thermodynamic phase of the REEDS code. In so doing, the meteorology, energy, and potential temperature factors are calculated. Then the range from the pad of the cloud center, the cloud transport direction, stabilization height, and surface mixing layer depth are calculated. These are necessary for the formation of one of the more significant calculations -- the wind azimuth uncertainty. This parameter is an important driver of the turbulent diffusion process. Once the surface mixing layer height is estimated and optimally refined, the RMTPS module is called to plot the pertinent geometric relationships and display the input meteorology and calculated potential temperature. The program is then ready for the kinetic phase of the calculation.

Data File Reads. The prompt for the surface layer height is transferred from the RCLDS question file, "?RCLDS" (Figure 4), to a holding array via the DREAD subroutine. The common disk file "=REEDS," is then read, some local variables and display flags initialized, and the exhaust cloud print header is written to the printing device.

Useful Constant Terms for the Meteorological/Energy Factors. In preparation for the calculation of the stabilized height and cloud rise energy factor, two terms useful for the calculation at a meteorological factor for bouyant rise,

$$GT = \frac{9.8}{T + 273.16}$$

and

$$\frac{.05129\ (T - 273.16)\ P_o\ \gamma}{Qc_1}$$

are calculated.

Adiabatic or Stable Cloud Rise. Next, a loop is performed through the meteorological layers to simulate the cloud rise phenomenon, either adiabatic or stable. The criteria is the argument of the inverse cosine (ARCCOS) term in the bouyant cloud use equation. If the argument is less than (-1), then the rise is "stable," and the flag so indicating, IAS, is set to 1; otherwise, the rise is "adiabatic," and IAS is 0. IAS is initialized in each layer to 1 ("stable" cloud rise).

Cloud Stabilization and Criteria for Surface Mixing Layer Height Selection. The slope (derivative with respect to altitude) of potential temperature (i.e., the "Lapse Rate"), the wind speed, and wind direction, in each layer are then calculated. The reader may recall the optional methods of calculating the lapse rate set up by the executive module, REEDS. If the least squares method (resulting in a lower cloud rise) is selected, a parameter IPTZ was set to 0. Otherwise, the alternate method of determining the lapse rate was chosen (IPTZ = 1), and is now calculated. In either case, a lapse rate is determined, and the energy factor calculated:

$$\alpha = \frac{\alpha_c (Z_i - Z_o)}{\Delta t_R} ,$$

where Z_i and Z_o are the current and initial layer altitudes, and Δt_R is the deposition time for the launch vehicle as defined by the vehicle's time versus altitude profile. The stability factor, STAB, is calculated from either lapse rate, determining adiabatic (STAB $\leq$ 0) or stable (STAB > 0, $[1 - \frac{1}{2} \alpha \, STAB] < -1$) rise. If the rise is adiabatic, the rise time through the current layer is merely

$$\sqrt{\alpha} \ ;$$

otherwise it is

$$\left[\frac{\pi/2 - \tan^{-1}(C_3)}{\sqrt{STAB}} \right]$$

where

$$C_3 = \frac{[1 - 1/2 \, \alpha \cdot STAB]}{\sqrt{1 - [1 - 1/2 \, \alpha \cdot STAB]^2}}$$

In addition to determining cloud rise, the three slopes calculated above become input factors to a deterministic calculation of the estimated surface mixing layer height to be performed after the conclusion of the rise loop and final stabilization height calculation:

Stabilization Range and Transport Direction. As the cloud rises through the meteorological layers, the wind speed and direction change. This affects the cloud direction and resultant position, so these effects are determined. This results in a transport direction and a stabilization range increment. At the conclusion of this loop, a stabilization range and height are calculated. Later, the range and direction will be averaged over the rise to produce a net direction throughout the single surface mixing layer.

Mixing Layer Height. Subroutine FTOP is then called to estimate the height of the surface mixing layer. To do this, the slopes of potential temperature, wind speed, and wind direction versus time are examined for "significant" changes. Just what is significant is somewhat subjective. Currently, three slope change magnitudes are stored in a data statement. They are:

$$\left| \Delta \frac{d(PT)}{dZ} \right| = 0.03$$

$$\left| \Delta \frac{du}{dz} \right| = 0.015$$

$$\left| \Delta \frac{d\theta}{dZ} \right| = 0.015$$

for potential temperature, wind speed, and wind direction, respectively. When these are exceeded by the respective slope changes, the changes are considered "significant."

If no significant changes are found, the mixing layer height is set to twice the stabilization height. If only one of the variables has a significant change, the altitude of that change is selected. If there are more than one, they are sorted in descending order of magnitude. If there is an altitude at which all three variables have significant changes, this altitude is chosen. If, instead, there are two with coincident altitudes of significance, that altitude is chosen. If there are multiple multi-variable altitudes of significance, or several different altitudes which are non-coincident, then the largest change is selected. The following list summarizes the selection priority:

1) The largest triple variable coincident significant change.

2) The largest double variable coincident significant change.

3) The largest potential temperature significant change.

4) The altitude of the single significant change.

5) Twice the stabilization altitude.

<u>Wind Azimuth Uncertainty</u>. A very important driver of the turbulent diffusion process is the wind azimuth uncertainty (σ_{Az}). This variable is computed by the subroutine "RSGAZ," using a theoretical and deterministic method developed by Panofsky from similarity theory. The alternative to a method such as this would be a statistical method. However, with soundings often used as input meteorological sources, resolution at near-surface altitudes may not be good. This is greatly improved with tower data. Nevertheless, the similarity theory-based deterministic method is an important contribution for predictive application to single layer turbulent diffusion.

The value of σ_{Az} depends on the value selected from FTOP for the top of the surface mixing layer. That height is not final (i.e., it may be changed by the user if the Research or Operational mode is used, so final calculation of σ_{Az} occurs when the mixing layer height is accepted. The form used for σ_{Az} is

$$\sigma_{Az} = \frac{\sigma_v}{\bar{u}} = \frac{kf(B)}{\ln(z/z_0) - \psi(R_I)},$$

σ_v	=	standard deviation of lateral component of turbulence (m/sec)
$\bar{U}$	=	mean wind speed (m/sec)
k	=	Von Karman's constant (=.4)
f(B)	=	Piecewise linear fit to experimentally measured values of $\sigma v/_u$ as a function of the non-dimensional stability ratio, B
z	=	altitude (m)
z_0	=	roughness length (m)
ψ(RI)	=	a universal function of the Richardson number.

This relation was derived by Lettau (unpublished) and Panofsky (1963), serving both stable and adiabatic atmospheres (International Journal of Air and Water Pollution, Vol. 9, 1965, Panofsky and Prusad on "Similarity Theory and Diffusion"). The non-dimensional stability ratio, B, is given by

$$B = \frac{g}{T}\frac{\bar{Z}^2}{\bar{U}^2}\frac{\Delta(P_T)}{\Delta z}$$

where

g	=	gravitational acceleration (m/sec^2)
$\bar{Z}$	=	the geometric mean of layer top and bottom (m)
$\bar{U}$	=	mean wind speed (this layer) (m/sec)
T	=	absolute temperature (^{o}K)
$\Delta P/_{\Delta z}$	=	lapse rate (vertical gradient of potential temperature).

Depending on the value of B, f(B) is selected from

$f(B) = a + b\,(B + c)$, where

A = 2.7, b = c = o	for B < -.008
= 2.7, b = 112, c = .008	-.008 ≤ B < -.00175
= 3.4, b = 725.5, c = .00175	-.00175 ≤ B < .0008
= 1.55, b = 38.04, c = -.0008	.0008 ≤ B < .029
= 2.35, b = 5.43, c = .029	.029 ≤ B

The Richardson number for ψ(RI) cannot be calculated from its definition (normalized lapse rate per unit vertical wind speed gradient squared), because sufficiently accurate measurements of the vertical wind speed gradient do not normally exist. It can be transcendentally obtained from:

$$R_I = B\left[\frac{\ln(z/z_o) - \psi(R_I)}{\phi(R_I)}\right]^2$$

where (RI) = $(1\text{-}16\,R_1)^{-1/4}$, unstable conditions

$(1\text{-}7RI)^{-1}$, stable conditions

(another universal function),

and

$$\psi(RI) = \ln \frac{1+X}{2} + \ln \frac{1+X^2}{2} + \frac{\pi}{2} - 2\,\mathrm{Tan}^{-1}(x), \text{ unstable condition}$$

$$= 7\,R_I(1-7RI)^{-1}, \text{ stable conditions}$$

where $X = (1-16RI)^{1/4}$.

For stable conditions,

$$\sqrt{RI} = (-1 + \sqrt{1 + 4\,A_1A_3}\,)/2A_1$$

where $A_1 = 7 \quad 8k$, and $A_3 = \; - \; B\,(k-1)$, for $k = \ln Z/Z_0$.

For unstable conditions,

$$\frac{(1-x)^4}{16\,X^2[\ln z/z_0 + .50864 - 2\ln(1+x) - \ln(1+x^2) + 2\tan^{-1}x]} - B = 0$$

is solved transcendentally for X, which yields RI.

In RSGAZ, the transcendental function statement and its necessary derivative precede the active statements (follow the detailed flow chart in Figure 2-5). The critical data (transferred via common) is selected from the input layers as a result of the input arguments passed. The potential temperature gradient (Lapse Rate) is then calculated at the indicated altitudes, followed by the Richardson number for either stable or unstable conditions. The appropriate value of f(B) is then obtained for the final calculation of σ_{Az}. The value of σ_{Az} returned is constrained between the limits of 7 degrees and 30 degrees, to keep resultant uncertainties physically realizable.

Interactive Alteration of the Surface Mixing Layer Height and the Wind Azimuth Uncertainty. Once the mixing layer height and subsequent wind azimuth uncertainty are estimated by the program, the user is given the opportunity to change each, if the RESEARCH or OPERATIONAL mode has been chosen. In the production mode, the user must enter both. The user is first prompted to use the estimated layer height or an altered value. If the altered value choice is

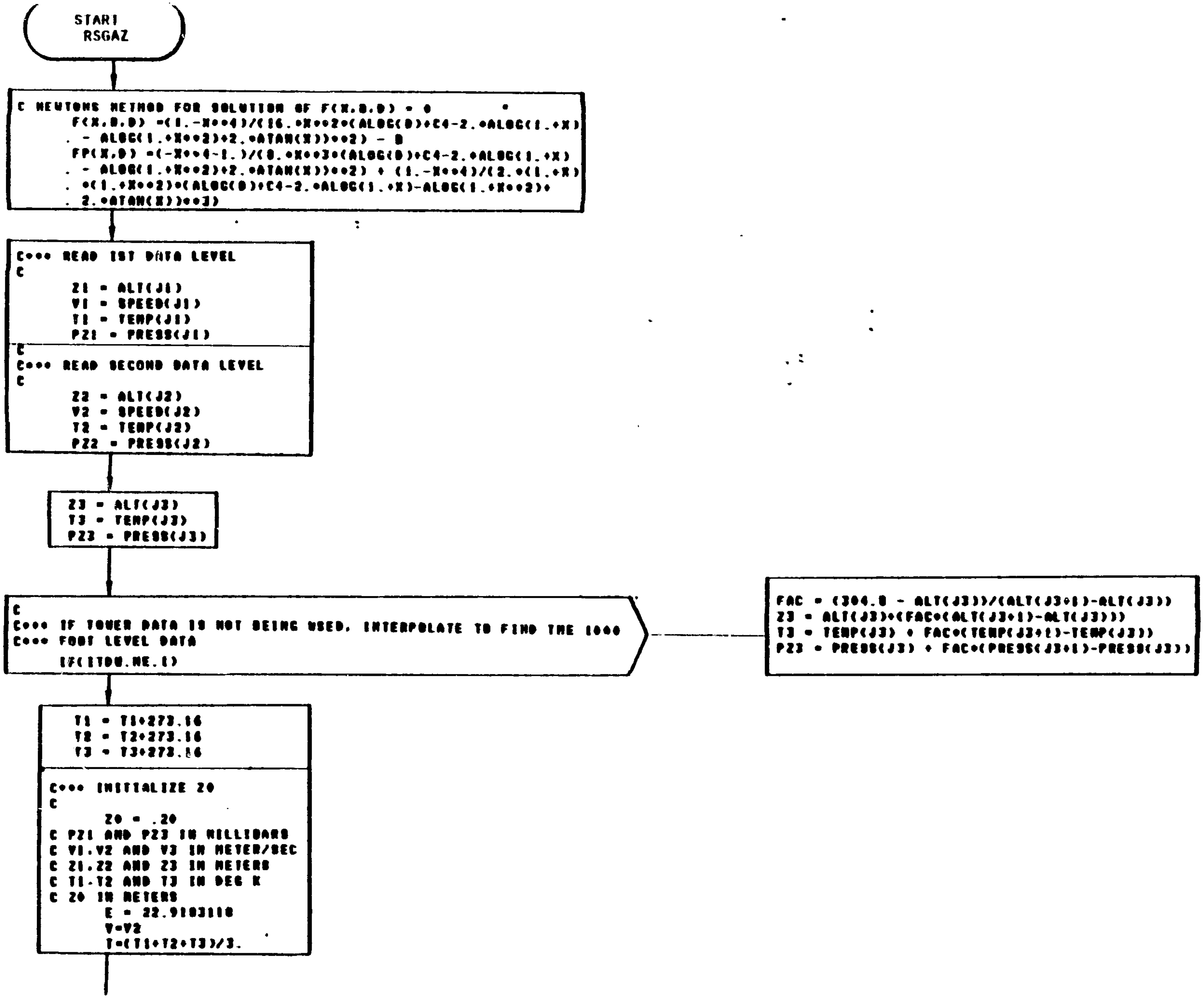


Figure 2-5. Detailed Structure Flow Diagram of Subroutine RSGAZ

ORIGINAL PAGE IS
OF POOR QUALITY

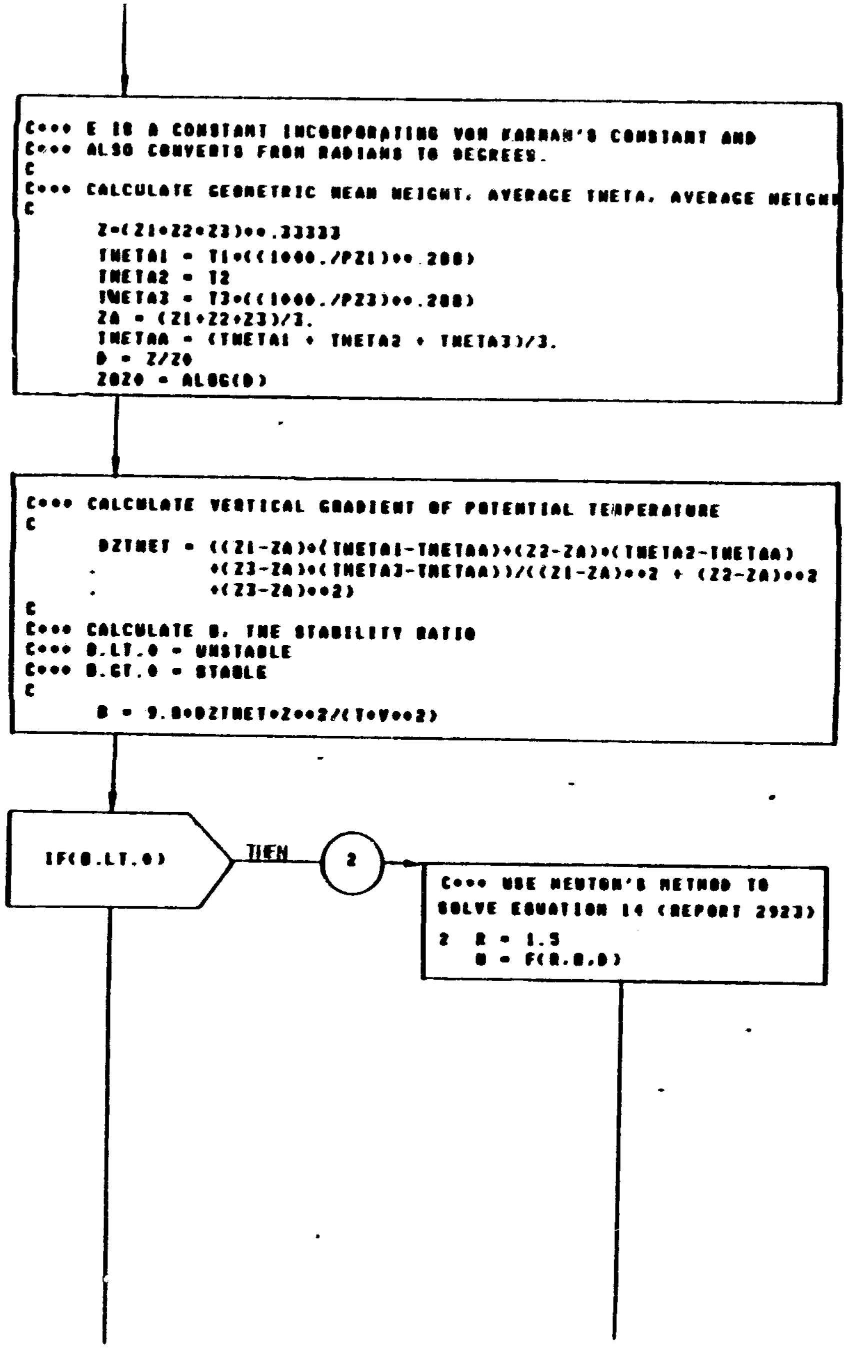


Figure 2-5. Detailed Structure Flow Diagram of Subroutine RSGAZ (continued)

ORIGINAL PAGE IS
OF POOR QUALITY

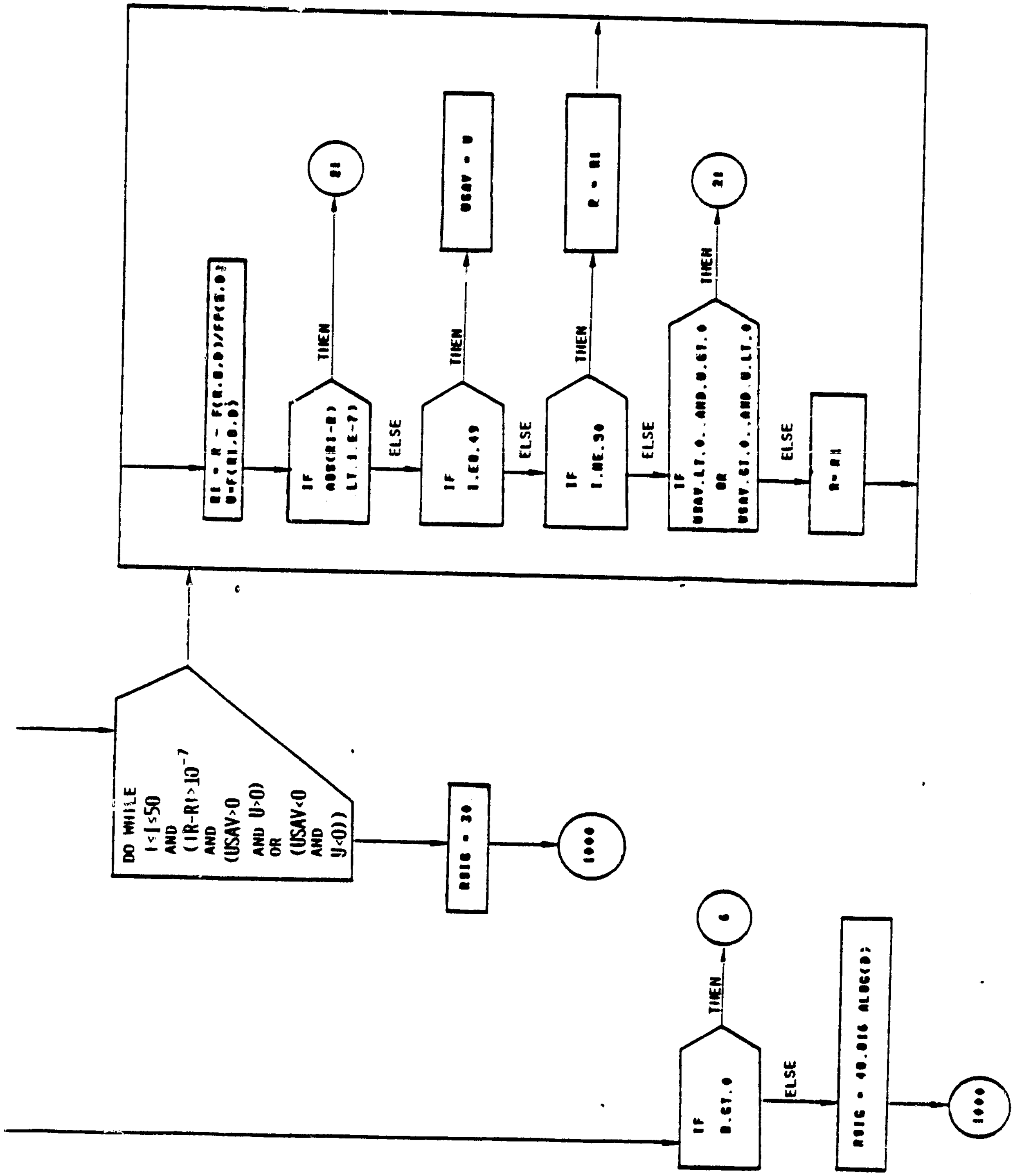


Figure 2-5. Detailed Structure Flow Diagram of Subroutine RSGAZ (continued)

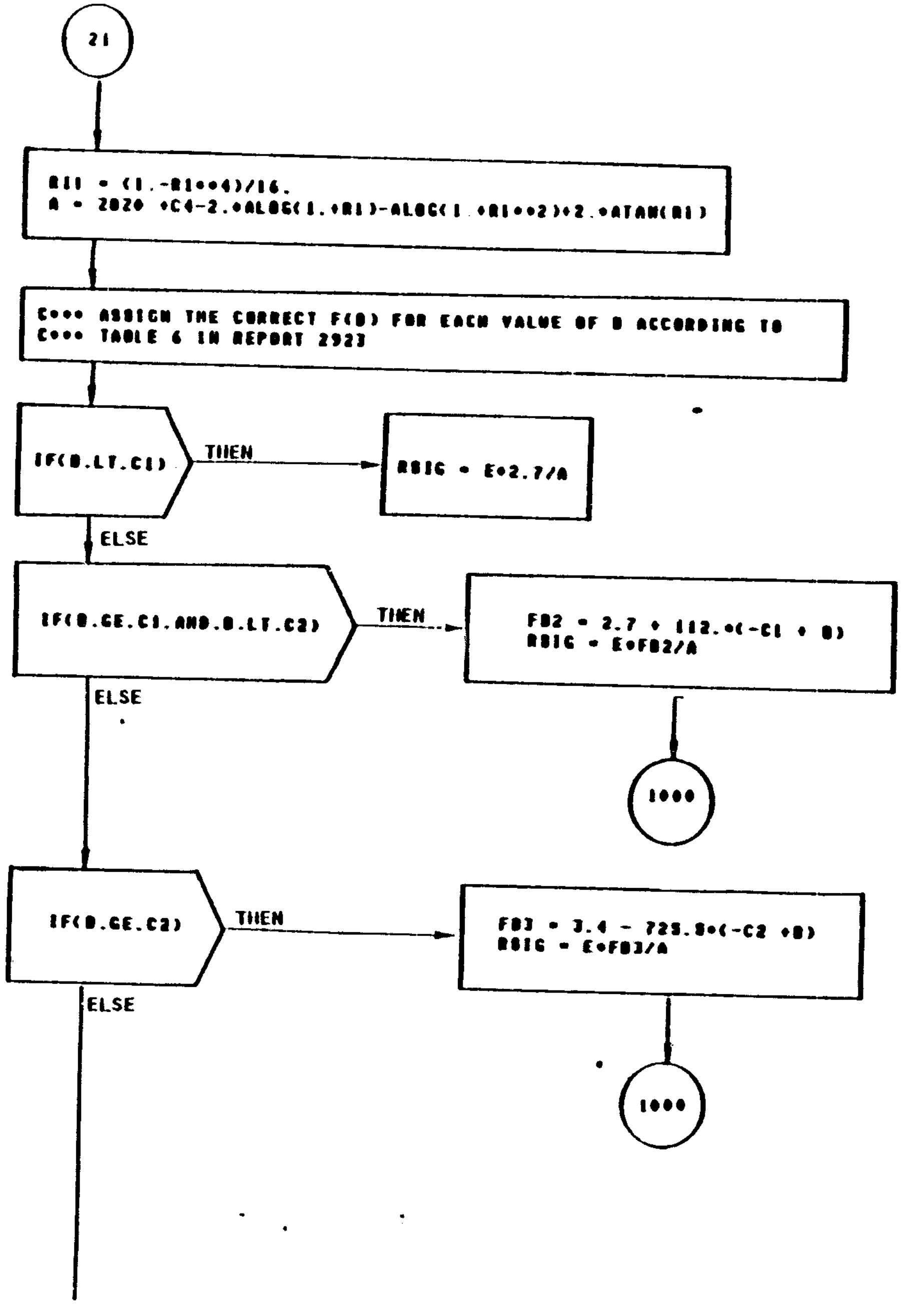


Figure 2-5. Detailed Structure Flow Daigram of Subroutine RSGAZ (continued)

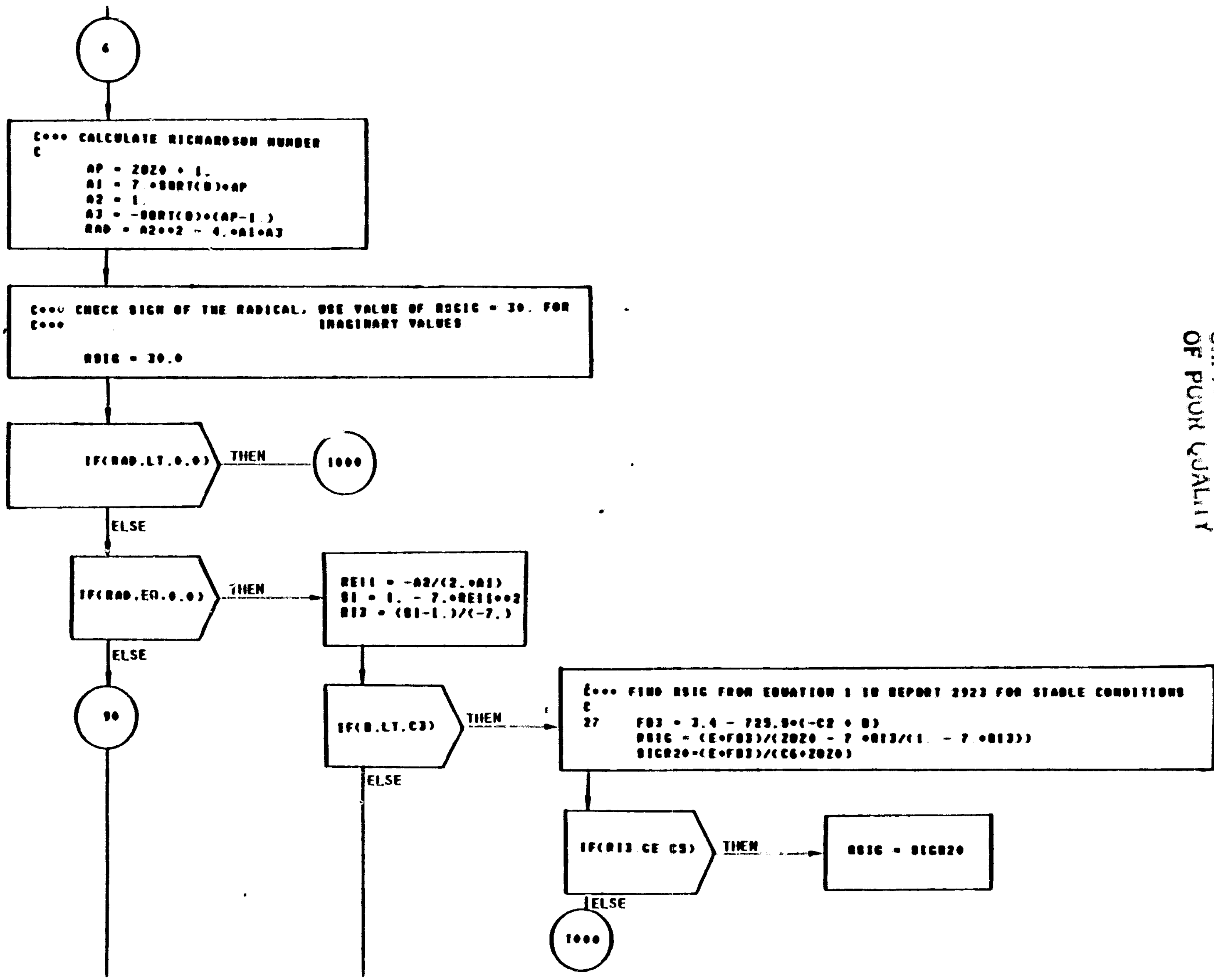


Figure 2-5. Detailed Structure Flow Diagram of Subroutine RSGAZ (continued)

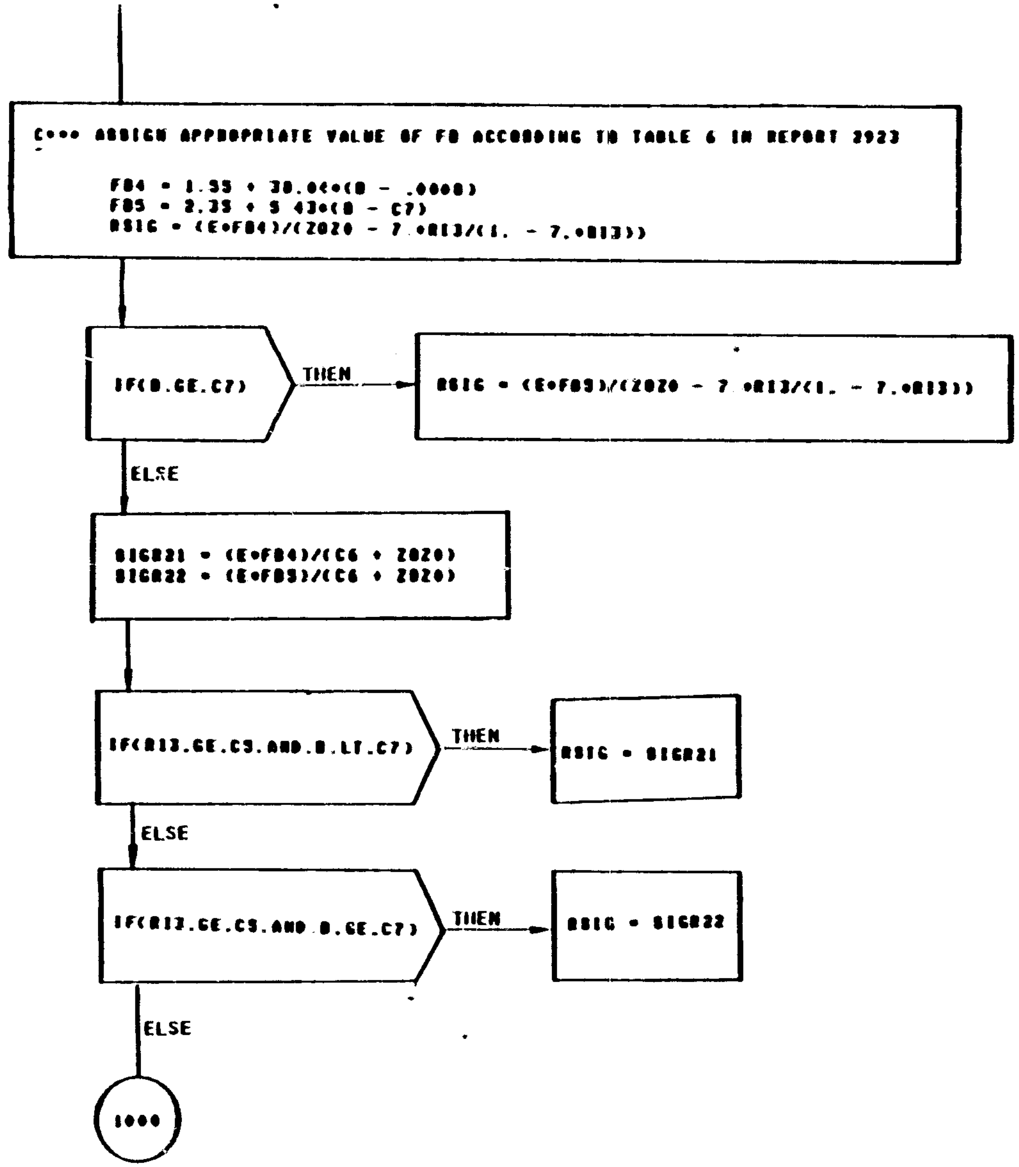


Figure 2-5. Detailed Structure Flow Diagram of Subroutine RSGAZ (continued)

ORIGINAL PAGE IS
OF POOR QUALITY

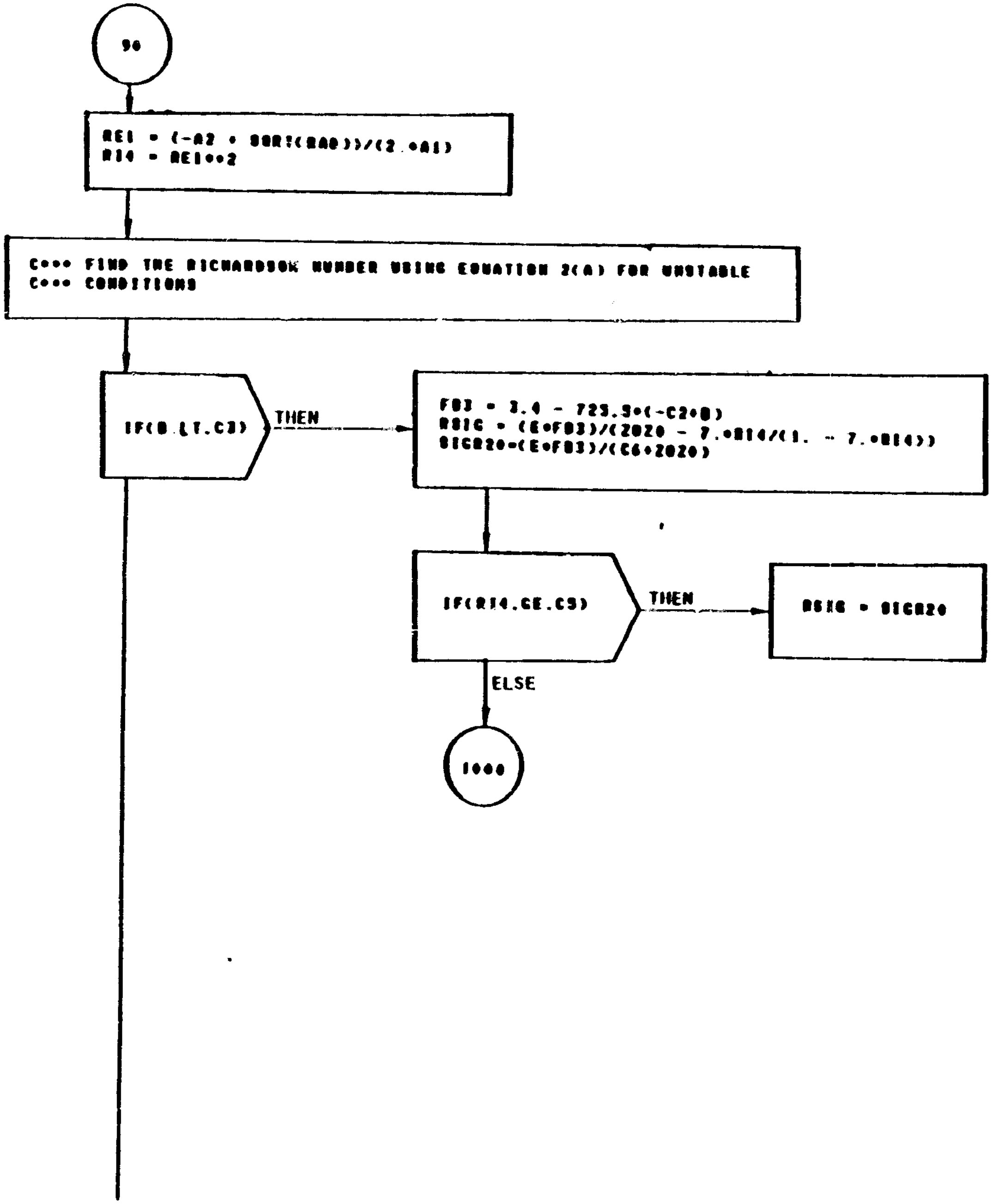


Figure 2-5. Detailed Structure Flow Diagram of Subroutine RSGAZ (continued)

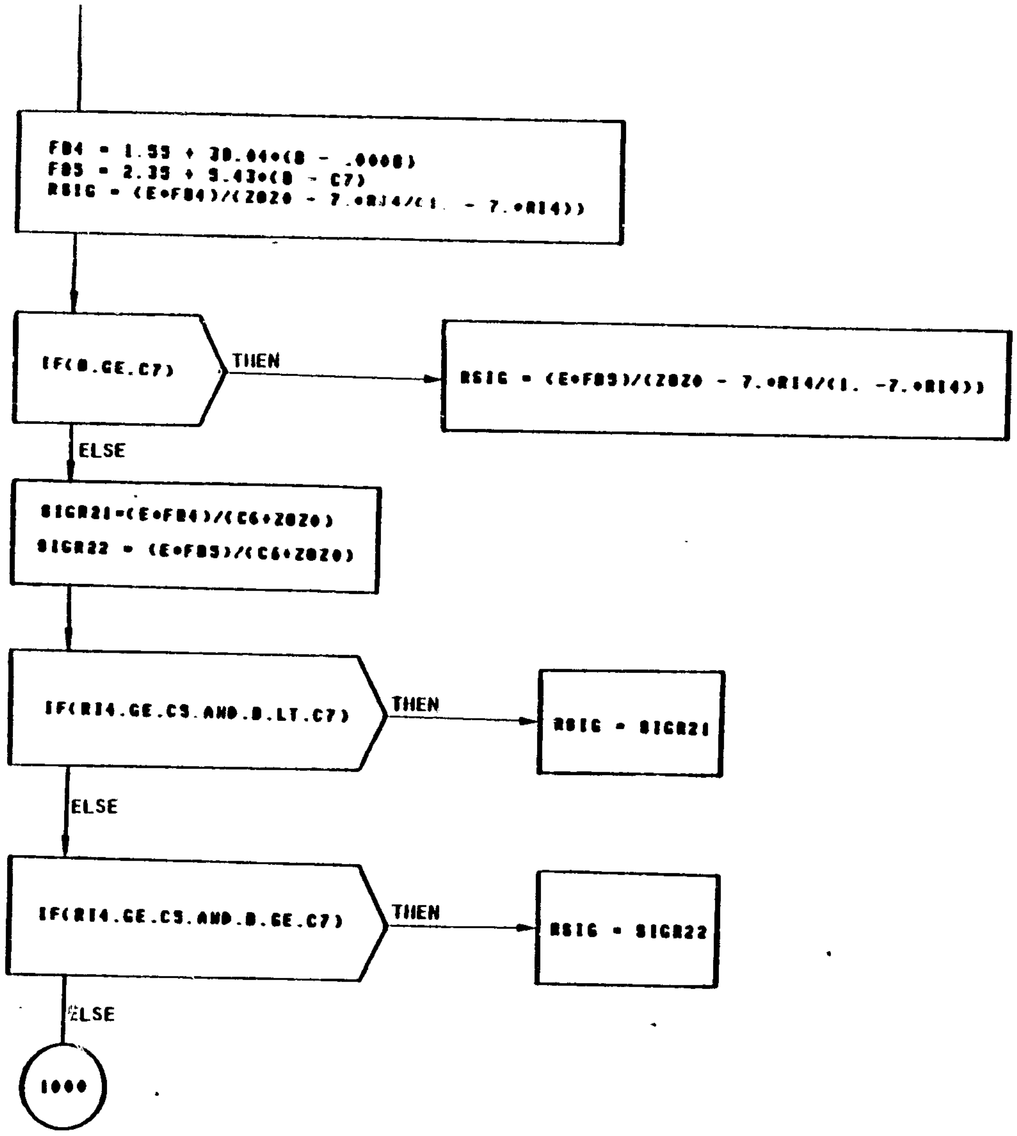


Figure 2-5. Detailed Structure Flow Diagram of Subroutine RSGAZ (continued)

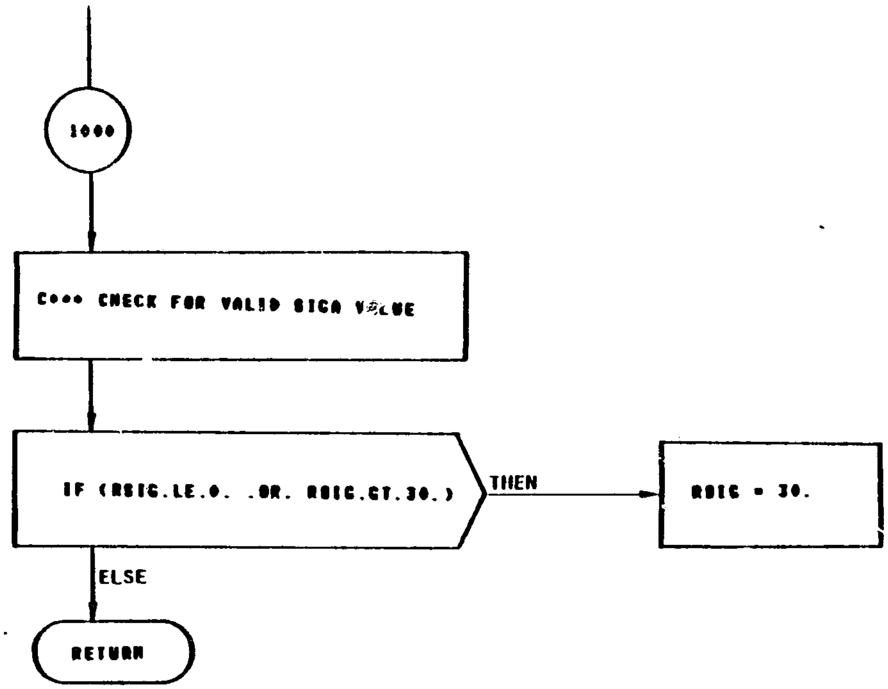


Figure 2-5. Detailed Structure Flow Diagram of Subroutine RSGAZ (concluded)

selected (via entry of an integer > 8) an interactive selection of the layer height is assumed. In either case, the common data are written to "*REEDS," and the meteorological data plotting routine "RMTPS" is called. Upon conclusion of RMTPS, control is returned to RCLDS. If met plots only were desired, the call to "RMTPS" is followed by a program STOP for immediate termination of execution. Otherwise, the height selected during "RMTPS" becomes the estimated mixing layer height for the balance of the calculations.

Contaminant Source Strength. Calculation of the mixing layer height determines the source strength of HCL (from which the other contaminant strengths are determined) in that it determines the maximum altitude over which the rocket vehicle effluent production algorithm may operate; contaminants dumped above the surface mixing layer are not permitted to diffuse into the layer.

Final Transport Speeds Direction, Wind Azimuth Uncertainty and Stabilization.. Definition of the mixing layer height also permits correct averaging of transport speed and direction over the layers within the mixing layer, so this is accomplished and the results written on the selected printing device. The user is then given the chance to change the wind azimuth uncertainty, if the RESEARCH or OPERATIONAL mode is active. If not "N" for No, the user enters the uncertainty desired in wind azimuth (degrees) and σ_{Az} is printed on the printing device. Since the mixing layer height selected (H_T) may have resulted in part of the cloud being above the mixing layer top, and therefore being discarded, the "effective" stabilization heat (H_{SE}) of a "revised" cloud containing only that portion within the mixing layer must be calculated. This is done according to

$$H_{SE} = \begin{cases} \dfrac{H_T + H_S - R_C}{2}, & H_T - H_S \leq R_C \\ \dfrac{H_T}{2}, & |H_T - H_S| > R_C \end{cases}$$

where H_S is the stabilization height and R_C the cloud radius.

Similarly, the vertical extent from cloud center (σ_{Zo}) is calculated as

$$\sigma_{zo} = \frac{H_T - H_S + R_C}{4.3}, \quad H_S + R_C \geq H_T, \quad \sigma_{zo} > 0$$

$$= \sigma_{Xo} = \frac{\gamma_M H_S}{4.30}, \quad H_S + R_C < H_T \text{ or } \sigma_{zo} \leq 0$$

where σ_{Xo} is the initial cloud radius.

The stabilization height is written to the printing device, the common data are written to the disk file, and the effluent concentration module "RCONS" is called. Following this, a program STOP is generated, returning control to the "REEDS" executive module.

2.1.3 The RMTPS Meteorological Data Plotting Module

The "RMTPS" module generates a plot of the Meteorological Profile Variables of wind speed and wind direction as a function of altitude, and the initial and effective stabilized cloud shape. The mixing layer is also indicated on the plot and can be interactively determined from user input. As an additional feature, the sounding (or other) sampling points are indicated as additional tic marks on the ordinate altitude scale. The "RMTPS" module also will draw the met plot forms if the user has selected that option in "REEDS."

Forms Generation. After the common disk file is read to load the common data, the RMTPS module checks for the forms switch, MFORM. If on (MFORM = 1), the form is generated; the first two hundred active lines (or so) are dedicated to this purpose.

Met Plots. After the forms are generated, or when forms generation is bypassed, the MONLY common parameter (OPTION control pseudo-block) is checked to see if MET plots are skipped altogether (or in a forms generation run or in a repeat run designed to prevent needless replotting of the MET-related data). The wind direction data is rotated to conform to the preset site orientation, and is scaled to plot units. The launch data and time, site data, surface pressure and density, stabilization height and wind azimuth uncertainty are all encoded into ASCII character strings and written by the plotter onto the MET plot. This is followed by the dry temperature, potential temperature, wind speed, and wind direction (all surface values), which are likewise encoded and written by the plotter. In each case, the encoding is accomplished by the combination of "CODE" calls and subsequent WRITE to the character string, IALPHA. The character string is then written at the designated plotter location via the "CHAR" subroutine.

Curve Drawing. Actual curve drawing then proceeds by pulling the array values for the various arrays out and using them as data for the "CHAR" calls. This results in the indicated symbol from the symbol array ICRVT being plotted. In the case of the line plots, either "LINE" (for solid lines) or "DLINE" for dashed lines are used for curve generation on the plotter.

Axis/Tic Marks. Curve drawing is then followed by axis generation. This is accomplished with simpler calls to LINE, whether for axes or for tic marks.

Cloud Drawing. The cloud schematic is drawn by the "CLOUD" subroutine in the lower right quadrant of the plot. Relative distances with respect to mixing layer are kept to approximate scales.

Mixing Layer Height. At the appropriate height on the altitude scale (ordinate), the estimated mixing layer is drawn with the "LINE" routine. The "MOVEM" subroutine is then called to prompt the user to accept or replace the indicated height. The choices are:

"CO"	for continue
"UP"	for move up one met level
"DO"	for move down one met level

After each prompting, the new line is drawn in. When it is finally accepted by entering "CO," the line is labeled by the encoded layer height (meters).

Effective Cloud. Some cloud parameters (radius, height, etc.), calculated in RCLDS but not saved, are recalculated. This permits redrawing the cloud schematic with the "effective" parameter. This time, however, the "LINE" routine is used with the cloud extent data to produce a shaded ellipse superimposed over the original cloud outline. After this operation, the common data are written to the disk file and a program "STOP" is generated, returning control to "RCLDS."

2.1.4 The RCONS Centerline Concentration Module

The RCONS concentration module computes the range and azimuth of the cloud center relative to the pad, calculates and plots the centerline dose and concentration of the major hazardous effluents, computes the off-centerline dose and concentration for specific locations, and calls the "RISOS" dosage and concentration isopleth plotting module. The "RCONS"/"RISOS" combination comprises the "kinematic phase" of the effluent diffusion problem. Refer to Figure 2-3 for the RCONS top-level flow.

The turbulent diffusion problem may be described by the following diffusion equation:

$$\frac{2\chi(\bar{r},t)}{2t} + \langle\bar{v}\rangle \cdot \nabla\chi(\bar{r},t) = \nabla \cdot \left[\tilde{K}(\bar{r},t,p,T) \cdot \nabla\chi(\bar{r},t) \right]$$

where $(\vec{r},t)$ is the scaler concentration of the diffusing effluents, $\langle\bar{v}\rangle$ the expected value of the wind velocity and $\tilde{K}$ (r,t,P,T) the diffusion diagonal tensor which is a function of temperature, pressure, position and time. The separation of the problem into two (2) distinct phases, a thermodynamic phase resulting in approximate equilibrium with the ambient environment and a kinematic phase, with diffusion from the thermodynamically stable state, facilitates the linearization of the equation and subsequent separation of variables solution. We have already looked at the first phase (RCLDS), and now wish to examine the latter.

The first phase left the exhaust cloud initially spherical in shape with the cloud center at a stabilized altitude. Then, the revision of the cloud occurred to account for surface mixing layer effects. Now, the ellipsoidal expansion of the revised cloud occurs. The generalized dosage equation driving this expansion is composed of four terms:

$$\text{DOSAGE} = \begin{pmatrix}\text{PEAK}\\ \text{DOSAGE}\end{pmatrix} \times \begin{pmatrix}\text{LATERAL}\\ \text{TERM}\end{pmatrix} \times \begin{pmatrix}\text{VERTICAL}\\ \text{TERM}\end{pmatrix} \times \begin{pmatrix}\text{DEPLETION}\\ \text{TERM}\end{pmatrix}$$

In subsequent modeling, the depletion of material from the cloud by gravitational settling and rain scavenging is included, but in the current REEDS model, no depletion of this type is included. Therefore, the driving dosage algorithm is:

$$D\left\{x, y, z_B < z < z_T\right\} = \frac{Q_M}{2\pi\ \sigma_y\ \sigma_z\ \bar{u}} \left\{\exp\left[\frac{-y^2}{2\sigma_y^2}\right]\right\} \left(\exp\left[\frac{-(H-z)^2}{2\sigma_z^2}\right]\right.$$

$$+ \exp\left[\frac{-(H - 2z_B + z)^2}{2\sigma_z^2}\right] + \sum_{i=1}^{\infty} \left\{\exp\left[\frac{-\left[2i\left(z_T - z_B\right) - \left(H - 2z_B + z\right)\right]^2}{2\sigma_z^2}\right]\right.$$

$$+ \exp\left[\frac{-\left[2i\left(z_T - z_B\right) + (H - z)\right]^2}{2\sigma_z^2}\right] + \exp\left[\frac{-\left[2i\left(z_T - z_B\right) - (H - z)\right]^2}{2\sigma_z^2}\right]$$

$$\left.\left. + \exp\left[\frac{-\left[2i\left(z_T - z_B\right) + \left(H - 2z_B + z\right)\right]^2}{2\sigma_z^2}\right]\right\}\right)$$

where:

- Q_M = Source Strength (total mass in the indicated surface layer)

T,B subscripts indicate top and bottom of stabilized cloud, respectively.

- H = Height of Cloud Centroid
- Z = Height within layer ($z_B \leq z \leq z_T$)
- σ_y, σ_z = Standard Deviation of the Time Dependent Cloud Dimensions in the y and z directions, respectively.

Centerline values are obtained at $y = 0$, while surface values are obtained at $z = 0$. From the dosage, the concentration is formed by the expanding front as

$$\chi(x,y,z,t) = \frac{D(x,y,z,t)\ \bar{U}}{2\pi\ \sigma_x}$$

where $\overline{U}$ is the average wind speed, and σ_X is the along-wind cloud dimension. The balance of the kinematic phase modules are concerned with establishing positions and timings relative to the cloud center and time of maximum dosage/concentration, and the relative distribution of dosage/concentrations among the effluent constituents.

Centerline Dosages and Concentrations. After some input terminal initialization, the standard common disk file read, and some useful parameter initialization, the range down the cloud centerline is computed and expressed relative to the pad. The dosage algorithm "guts" exists in subroutine DFEXP, where the various sums in the above equation are computed. DFEXP is called with y = 0 off centerline distance to obtain the centerline maximum dose and concentration. The arguments of DFEXP are the surface mixing layer height and the concentration desired. DFEXP can then obtain the appropriate dimensions corresponding to that concentration. However, the specification of the number 1000 for concentration is a key that causes DFEXP to produce only values at the indicated range, and report the standard deviation of the cloud dimensions in the transverse direction.

Cloud Length and Arrival/Departure Time. Based on the calculations of cloud extent made by DFEXP, the arrival and departure times at the indicated range are computed and printed with the range and azimuth relative to the pad, at 250 meter increments along the cloud centerline.

RCONS User Prompts. Most prompts for RCONS user action are stored in the RCONS question file "?RCONS," and are accessed via the DREAD subroutine.

Centerline Plots. If the current operational mode is not production, the user is then prompted to decide whether or not to produce actual plots of the centerline dosage and concentration. If the response is not "N" for no, subroutine CPLOT is called with the centerline range array generated above. The plot is generated with a pre-set scale factor in order to coincide with traditionally generated forms. The default response to the centerline plot prompt is "YES."

Maximum Dosage/Concentration. The maximum dosage and concentration will occur along the cloud centerline. The point at ground level where this occurs had been calculated and saved during the above centerline dosage/concen tration calculation. They are now written out on the printing device selected with pad-relative range and azimuth.

Off-Centerline Concentrations/Dosages. If the current operating mode is PRODUCTION, the run is effectively complete and transfer of control is to the end of RCONS, and ultimately to the REEDS executive module for termination or restart. If the current mode is not PRODUCTION, however, the user is prompted for "off-centerline dosages/concentrations." If the answer is not "N" for "No," off-centerline concentrations/dosages are assumed to be desired (which is the default case), and the ORIGIN subroutine is called to set the origin for the desired launch pad and map. Application of the REEDS code to other sites can be accomplished by appropriately modifying the vehicle/pad association data in the ORIGIN subroutine. From ORIGIN, the user is prompted for the pad number and land/sea map combination. The choices are listed by the prompt, depending on the vehicle selected during REEDS input (certain vehicles imply certain pads). The user then inputs the code listed in the prompt which corresponds to the pad/map combination he desires. The user will now be prompted for pad-relative off-centerline ranges and azimuth. However, if the operating mode is RESEARCH, the user is first prompted (from the program) to enter a cloud centerline (stabilization-relative) range. This is then transformed to a pad-relative range and azimuth using the setup from ORIGIN. This is an aid in the verification of data from the centerline calculations and/or an additional centerline/off-centerline consistency check. In either RESEARCH or OPERATIONAL modes, the user is then prompted for the pad-relative range followed by the pad-relative azimuth. The entered range and azimuth are printed on the terminal display, and then on the printing device. The coordinates are plotted on the Isopleth plotting map requested by ORIGIN according to the setup resulting from the requested pad/map combination. The point is labeled by the ordinate number of the range/azimuth pair (first entry = 1, second entry = 2, etc.). Two additional points are then computed at the requested range, but the first at 10^{0} less azimuth and the second at 10^{0} more azimuth. All three range/azimuth pairs are then transformed to cloud center-relative coordinates, and DFEXP is called. The concentration and dosage is evaluated at each point and the $\pm 10^{0}$ data used to produce an uncertainty in the concentration/dosage for each effluent. The resulting data is printed on the printing device. The format is:

MATERIAL	CONCENTRATION (PPM)	± CONCENTRATION UNCERTAINTY	DOSAGE (PPM-SEC)	± DOSAGE UNCERTAINTY (PPM-SEC)

The above off-centerline calculation is repeated until a zero (0) range is entered.

Isopleth Plotting. The user is then prompted for Isopleth plotting. If the answer is not "N" for "No," the Isopleth plotting module is scheduled, otherwise RCONS is terminated and control returns to REEDS through RCLDS, for program termination/restart.

2.1.5 The RISOS Isopleth Plotting Module

The RISOS module essentially repeats off-centerline calculations using DFEXP, except it uses a clever series of calls resulting in iso-concentration contours. Refer to Figure 2.3 for RISOS top level flow.

Isopleth Initialization. The user terminal is first initialized and the common data read from disk via subroutine RWDIS. The ORIGIN subroutine is called again in case the call in RCONS was bypassed by rejection of the off-centerline option. This makes sure the plotting origin is established relative to the vehicle pad/map configuration desired.

Cloud Rise Ground Track. The cloud rise ground track from the pad to stabilization is then plotted by calculating the range of the cloud centroid from the time required to rise through each meteorological layer. The times for each layer were stored during the cloud rise module (RCLDS) execution. The average wind velocity in each layer is integrated over the stored time to provide the incremental range that is plotted. Since the wind velocity is measured "from" the source, the cloud transport direction must use a factor of π to properly orient its movement. The process is terminated when the layer containing the stabilized cloud center is reached. However, an additional segment is required, because of the use of the "effective" stabilization height, which replaced the stabilization height when placement of the surface mixing layer height severed part of the cloud (for diffusion purposes). The stabilization point is then labeled with a "+." Next, the point of maximum concentration at ground level, stored during RCONS, is labeled with an "@." Finally, the cloud track downwind from stabilization is plotted with a straight line using the transport direction from that point.

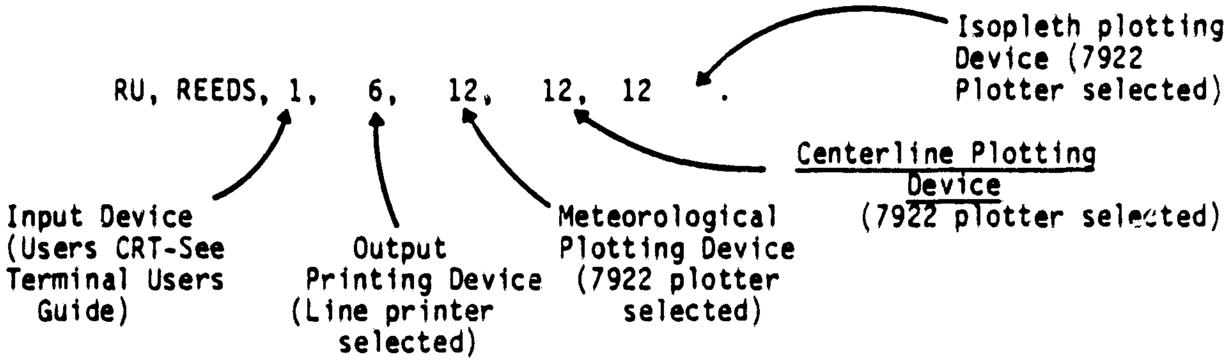
3.0 REEDS OPERATION

The basic form of storage for REEDS is in the form of source code files for the program/subroutine packages and ASCII data files for the input data. Ultimately, the source code files are compiled and thereby translated into machine language modules. The latter are loaded, SP'd, OF'd and RP'd (see terminal users manual). The "RU" command is then used to execute the RP'd modules while arguments are used to pass special information to the program. The compiled, loaded, and SP'd files are convenient to use, since they merely have to be RP'd prior to execution--a very quick process. One drawback is that there is little way of verifying that the RP'd module is the result of the source file as it appears. Also, since source files _must_ be saved anyway, the existence of a large compiled file may seem redundant, especially if not used often. Nevertheless, because of the lengthy loading process, the compiled files are often stored. Thus, for purposes of showing a run, _the compiled files are assumed to exist in the RP'd state_. Maintenance files have been built which automatically perform the operations leading to an RP'd program. It is the purpose of this section to explain actual REEDS execution.

3.1 THE TEST CASE

A test case was chosen, using the space shuttle as the vehicle and the 8 A.M. EDT 16 November 1969 rawinsonde as the meteorology. The file name of the latter data was changed to DATA01, just for convenience, and can be seen in Section 4.4 of this report. The CRT terminal used was equipped with a hardcopy device, making it possible to get intermittent copies of the screen to aid the narration.

Run Time Parameters. The REEDS system was designed to serve MSFC research needs and KSC operational needs. One difference between these is that KSC has three plotters, each one assigned to a major plot output. MSFC has _one plotter_ for all three plots. The allocation in the run statement of printing and plotting devices to logical units makes this possible. The user enters:



At this time the screen appears as in Figure 3-1. The default responses are indicated on the CRT screen when they appear by easily noted enhanced characters.

Operational Configuration. The next five (5) entries deal with configuring the logical path through the modules. The first prompt asks about the low altitude curve fit. We choose not to use the least squares fit for the potential temperature gradient and answer "No." We have renamed the 16 Nov 1969 rawinsonde file DATA01 for this run, and therefore select the default, "YES," response for the meteorological data file name (response-2). For the third input, the default OPERATIONAL mode is chosen, followed by a similar choice for Met Plot Form rejection (4th) and Met plot creation (5th). The printer output default ("Yes") is selected, and the screen appears as in Figure 3-2.

Launch Specific Parameters. The default launch time is the current time on the computer's system clock. It was decided to select the time rather than enter either a realistic or other time. Likewise, the default vehicle (space shuttle) is chosen as the launch vehicle. At this point, the screen appears as in Figure 3-3.

Initial Printed Output. At this point, the initial printer data becomes available, printing the header and meteorological sounding data as shown in Figure 3-4. The printout reflects all input entered to date, except the Met. Plot Form/Met. Plot configurational data.

Estimated Mixing Layer Height. The RCLDS cloud rise module has estimated the height of the top of the surface mixing layer, printing 2552 meters on the screen. The user is then prompted to raise or lower this height. The cloud rise has been calculated and stored up to this height, and RMTPS has been called.

```
RU,REEDS,1,6,12,12,12

     *****NASA/MSFC EXPERIMENTAL MULTILAYER DIFFU15 AUG 1981****
```

Figure 3-1. Code Initiation

```
 *****NASA/MSFC EXPERIMENTAL MULTILAYER DIFFU15 AUG 1981****
DO YOU WANT THE LOW ALT. CURVE FIT?  YES   NO  NO
LOW ALTITUDE CURVE FIT                                    NO

 USE DEFAULT COMMON DATA FILE (DATAØ1)   ?             YES

 RUN TYPE:                         OPERATIONAL

GENERATE THE MET PLOT FORM?                                   NO

MET PLOTS?                                             YES

GENERATE PRINTER OUTPUT?                               YES

DEFAULT LAUNCH TIME/DATE?  1833 CDT 15 AUG 1981        YES       NO
```

Figure 3-2. Configurational Paths Added

```
*****NASA/MSFC EXPERIMENTAL MULTILAYER DIFFU15 AUG 1981****
DO YOU WANT THE LOW ALT. CURVE FIT?  YES   NO  NO
LOW ALTITUDE CURVE FIT                                         NO

 USE DEFAULT COMMON DATA FILE (DATAØ1)   ?                  YES

 RUN TYPE:                                 OPERATIONAL

GENERATE THE MET PLOT FORM?                                          NO

MET PLOTS?                                                  YES

GENERATE PRINTER OUTPUT?                                    YES

DEFAULT LAUNCH TIME/DATE?  1833 CDT 15 AUG 1981             YES      NO

SELECT LAUNCH VEHICLE:                                         SHUTTLE=S

ESTIMATED TOP OF SURFACE LAYER(M) IS:  2552.4


MOVE TOP OF SURFACE LAYER:     UP     DOWN     CONTINUE
```

Figure 3-3. Launch Specific Data Added

```
********************************************************************************
********************************************************************************
************                                                        ************
************  NASA/MSFC REED CODE USING SINGLE LAYER MODEL 21 DEC 1979  ************
************                                                        ************
************                                                        ************
************                          REEDS                         ************
************                                                        ************
********************************************************************************
********************************************************************************
```

RUN 1 USING DATA FILE DATA01

SHUTTLE LAUNCH VEHICLE

PREDICTION TIME: 1007 CDT DATE: 24 AUG 1981

DATA FILE HEADER INFORMATION:
TEST NBR 03717 04834 0-24HR
RAWINSONDE RUN AN/GMD-4
CAPE CANAVERAL AFS, FLORIDA
ASCENT NBR 0543

TIME: 800 EDT DATE: 16 NOV 1969

```
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSS                                       SSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSS                SOUNDING               SSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSS                                       SSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
```

SURFACE DENSITY (GM/M**3): 1273.00

LAYER NO.	ALTITUDE (FEET)	(METERS)	DIRECTION (DEGREES)	SPEED (M/SEC)	TEMP	POT-TEMP	D P TEMP	PRESSURE (MILLIBARS)
					(DEGREES CENTRIGRADE)			
1	16	5	320	.4	7.1	4.96	23.2	1027.
2	735	224	10	.8	9.6	9.60	22.9	1000.
3	754	230	10	.8	0.7	9.76	21.7	999.3
4	2126	648	27	.7	6.5	10.66	16.6	950.0
5	2172	662	28	.7	6.4	10.69	15.3	948.5
6	3579	1091	7	.4	6.3	14.91	13.6	900.0
7	3629	1106	6	.3	6.3	15.04	13.6	898.6
8	5121	1561	316	.3	7.2	20.63	10.6	850.0
9	5960	1817	335	.2	7.7	23.78	8.7	824.2
10	6754	2059	9	.2	6.4	24.96	10.7	800.0
11	8374	2552	41	.1	4.9	28.51	7.6	753.5
12	8485	2586	41	.1	4.9	28.92	3.5	750.0

Figure 3-4. Initial Printer Data-Header and Meteorological Data

Initial Plotter Output. The meteorological data are plotted, since Met. Plots were selected; however, the form is omitted because a blank form already exists and is used. The blank form appears in Figure 3-5. After the Meteorological data are plotted, the plot appears as in Figure 3-6. The estimated mixing layer top is drawn, but not labeled, and the meteorological sounding data points are indicated by the additional irregularly spaced tic marks on the left ordinate. The program is left in an input mode, awaiting instructions as to mixing layer height modification.

Mixing Layer Height Modification. As a demonstration of the program options rather than a reflection of reality, the cloud is shown stabilized at 871 meters, and the mixing layer height is dropped four (4) times to 1106 meters. As can be seen in Figure 3-7, this is accomplished by entering four (4) consecutive "DO"'s for (DOWN) and finally a "CO" (for continue).

Final Cloud Rise. With the mixing layer finalized, the cloud rise is modified to reflect "severing" of part of the cloud by the boundary and subsequent recalculation of the stabilization height (to become the "effective" height) and total rise time. The resulting "effective" cloud is shown in Figure 3-8, and the cloud rise summary data printed on the printer is shown in Figure 3-9.

Final Wind Azimuth Uncertainty. The wind azimuth uncertainty is calculated from the mixing layer height and printed on the terminal display (Figure 3-7). The user is given a chance to modify this value. As the figure shows, the estimated value was accepted.

Centerline Concentrations and Dosages. The table of calculated centerline HCL dosages and concentrations are printed as shown in Figure 3-10. The user is prompted for centerline plots of concentration and dosages. These are desired (Figure 3-11) so a blank centerline form is loaded onto the plotter (Figure 3-12). The centerline dosage and concentration are plotted on the plotter, and the plot is labeled (Figure 3-13). The maximum ground concentration and dosage are then printed on the printer, as are the pad-relative coordinate (Figure 3-14). All dosages and concentrations are HCl dosages and concentrations. All other effluents are determined from HCl, the chief effluent.

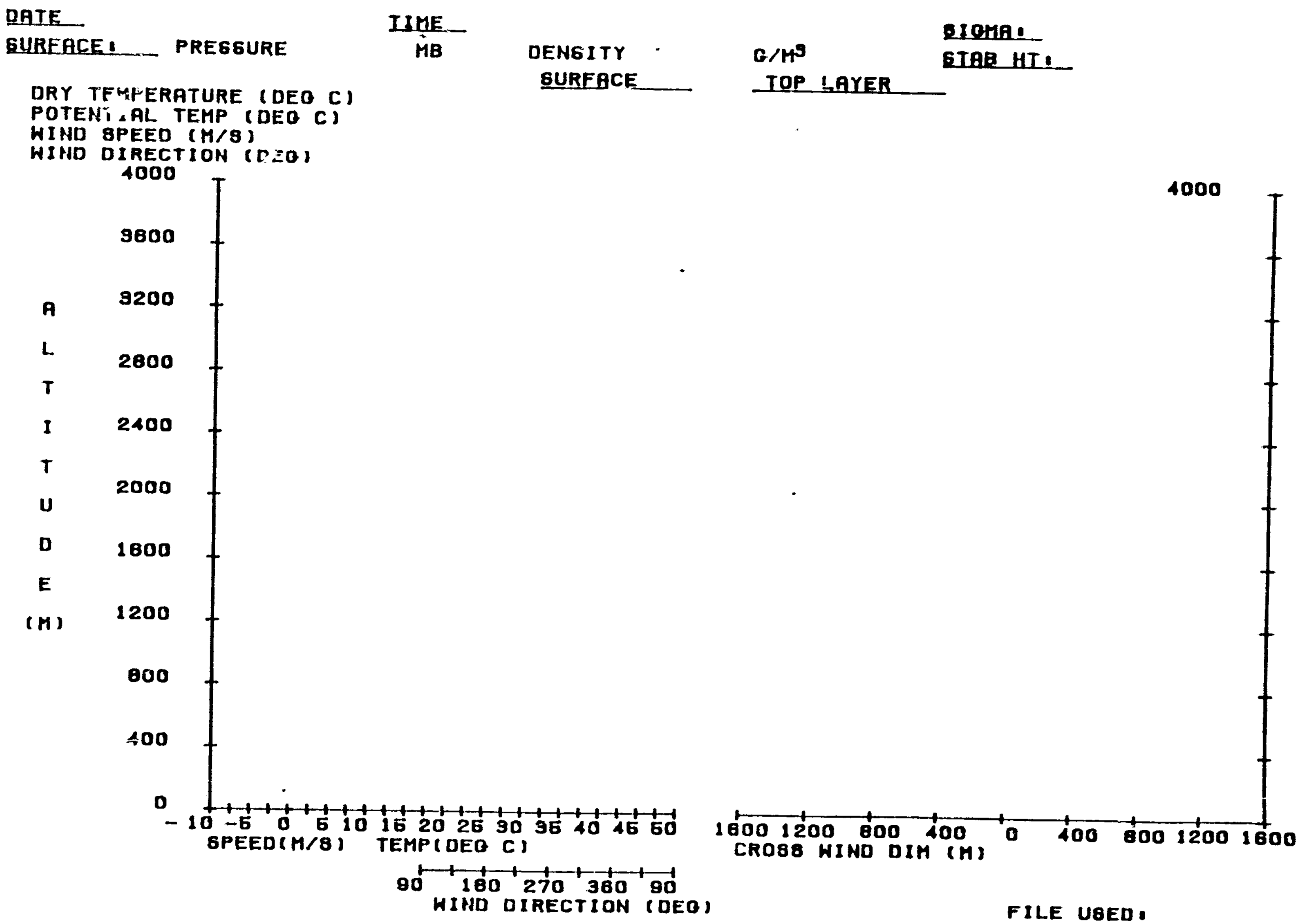


Figure 3-5. Blank Meteorological Form

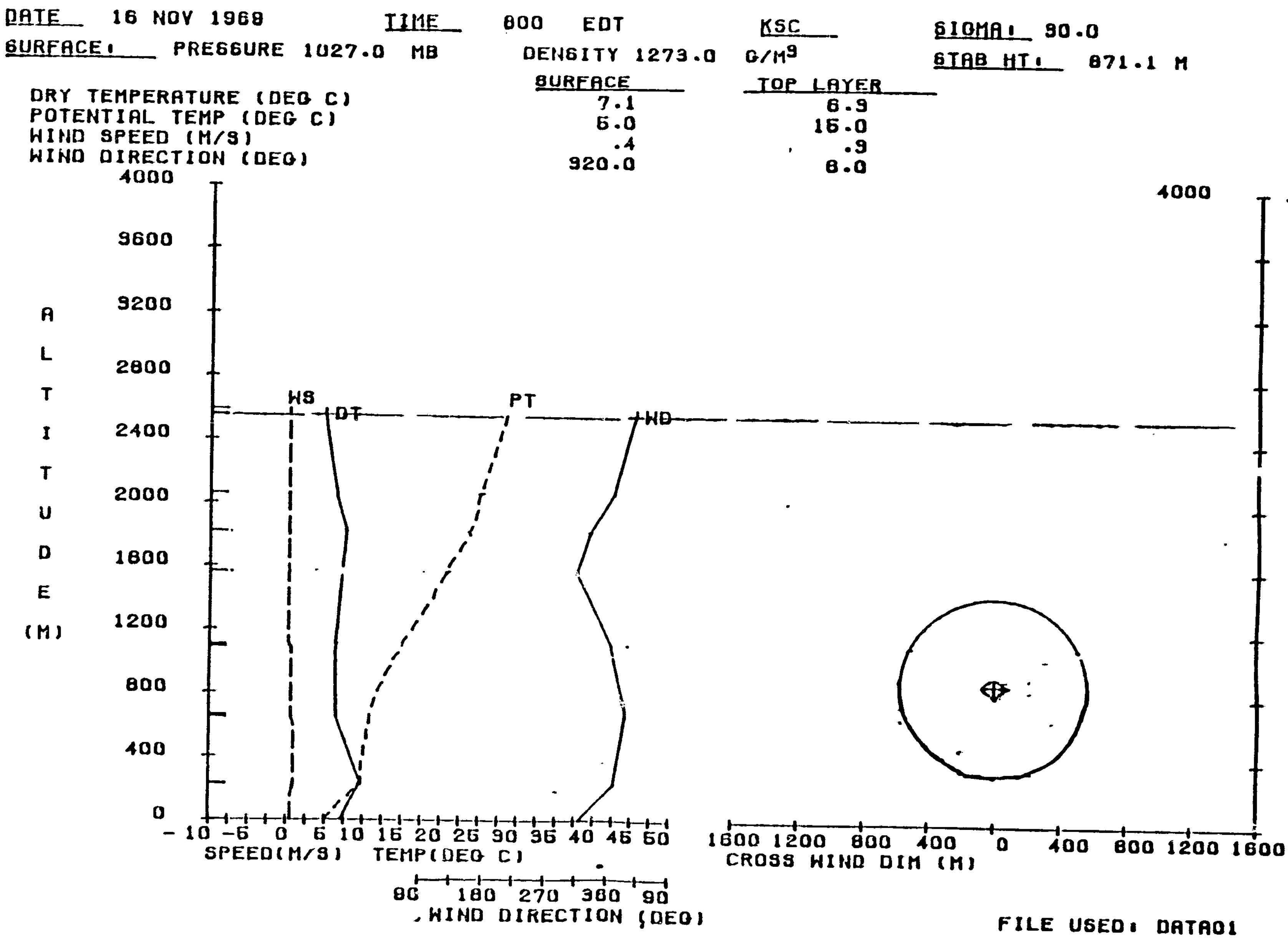


Figure 3-6. Meteorological and Initial Cloud Data Plotted with Estimated Mixing Layer Height

DO

DO
DO
DO
CO

CALCULATED TOP OF SURFACE TRANSPORT LAYER IS: 1106.12

USE A SIGMA FOR WIND AZIMUTH ANGLE OF: 30.0 YES NO

USE A SIGMA FOR WIND AZIMUTH ANGLE OF: 30.0 YES

CENTERLINE CONCENTRATION PLOT DESIRED ? YES NO

Figure 3-7. Modifying the Mixing Layer Height

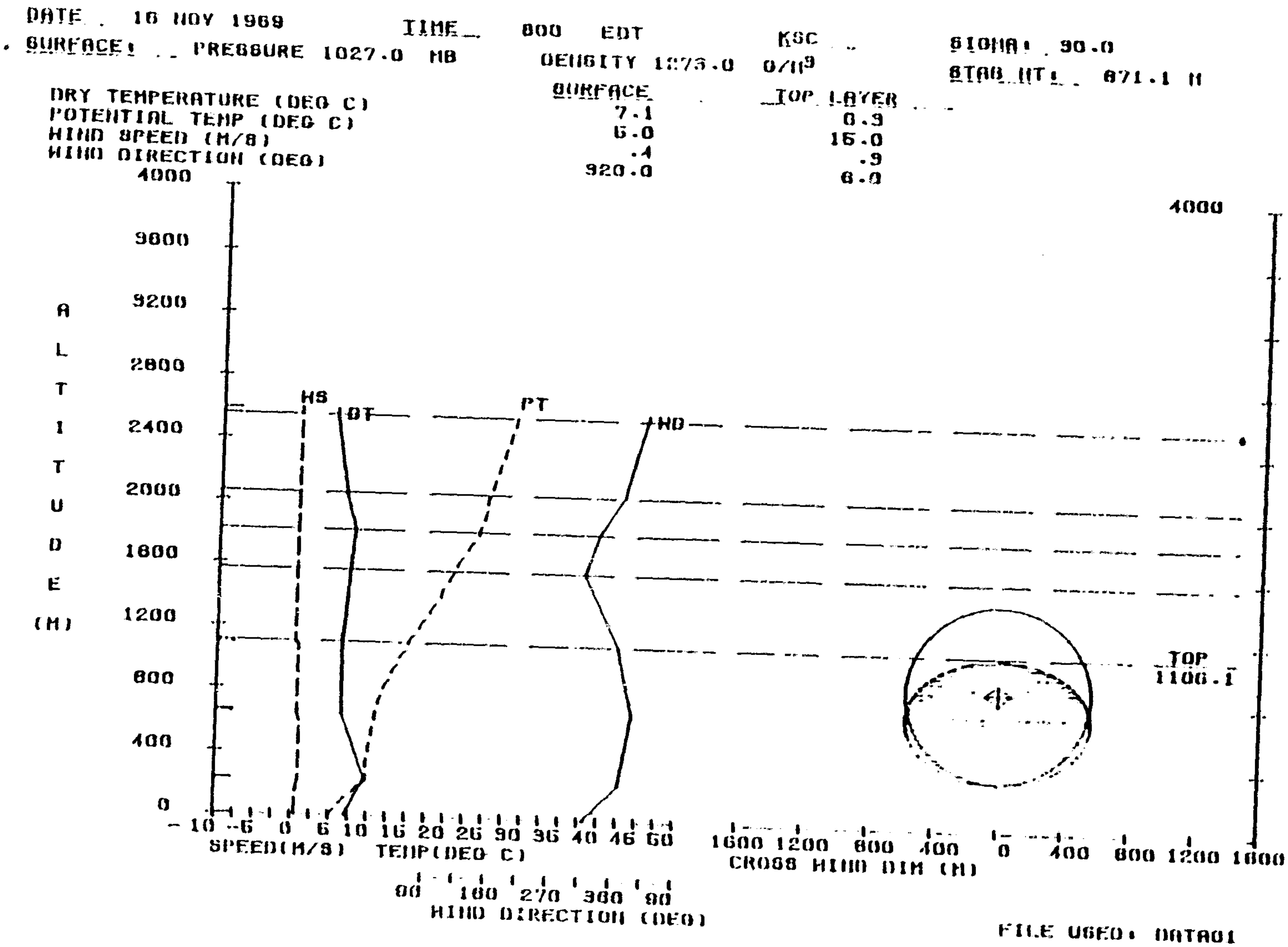


Figure 3-8. Modified Cloud - Showing Mixing Layer Height Effects

EXHAUST CLOUD

LEVEL	ALTITUDE (METERS)		RISE TIME (SECONDS)	RANGE (METERS)	DIRECTION (DEGREES)
2	224.0	STABLE	7.6	4.7	165.0
3	229.8	STABLE	8.0	5.0	166.5
4	648.0	STABLE	53.3	39.3	194.6
5	662.0	STABLE	55.5	40.8	195.1

```
****CLOUD STABILIZATION****
          HEIGHT(M): 1004.2
          STABILIZATION TIME AFTER LAUNCH(SEC): 174.0
          RANGE FROM PAD(M):  111.7
          DIRECTION FROM PAD(DEG):  195.3
PRESSURES ARE:    1027.00      1000.00     999.30
TBHT = .733812E+03


****TOP OF SURFACE LAYER METEOROLOGICAL PARAMETERS****
          HEIGHT(M): 1106.1
          WIND DIRECTION(DEG):  6.
          WIND SPEED(M/SEC):    .3


****DIFFUSION PARAMETERS****
          MEAN SPEED(M/SEC):    .7
          MEAN TRANSPORT DIRECTION(DEG):  191.3


SIGMA OF WIND AZIMUTH ANGLE, SIGAZ:  30.0


EFFECTIVE CLOUD HEIGHT (M):  733.8
```

Figure 3-9. Final Exhaust Cloud Rise Data

CLOUD CONCENTRATIONS AND DOSAGES

DISTANCE (METERS)	CONCENTRATION (PPM)	DOSAGE (PPM SEC)	TIME AFTER LAUNCH(SEC) START	FINISH	POSITION FROM PAD RANGE	AZ
0.0	.000	.057	-1787	2135.	111.7	195.3
1000.0	3.831	4379.79	-261.	3661.	1111.4	191.7
2000.0	5.126	5859.96	1264.	5186.	2111.4	191.6
3000.0	4.432	5066.42	2790.	6712.	3111.4	191.5
4000.0	3.618	4135.67	4316.	8233.	4111.4	191.5
5000.0	3.005	3435.43	5841.	9763.	5111.4	191.4
6000.0	2.564	2931.11	7367.	11289	6111.4	191.4
7000.0	2.235	2555.21	8893.	12815	7111.4	191.4
8000.0	1.981	2264.59	10418	14340	8111.4	191.4
9000.0	1.779	2033.24	11944	15866	9111.4	191.4
10000.0	1.614	1844.72	13470	17392	10111.4	191.4
11000.0	1.477	1688.16	14995	18917	11111.4	191.4
12000.0	1.361	1556.07	16521	20443	12111.4	191.4
13000.0	1.262	1443.14	18047	21969	13111.4	191.4
14000.0	1.177	1345.48	19572	23495	14111.4	191.4
15000.0	1.102	1260.19	21098	25020	15111.4	191.4
16000.0	1.037	1185.07	22624	26546	16111.4	191.4
17000.0	.978	1118.39	24150	28072	17111.4	191.4
18000.0	.926	1058.82	25675	29597	18111.4	191.4
19000.0	.879	1005.27	27201	31123	19111.4	191.4
20000.0	.837	956.869	28727	32649	20111.4	191.4

Figure 3-10. Centerline Hcl Concentrations/Dosages with Cloud Arrival/Departure Times and Pad-Relative Positions

DO

DO
DO
DO
CO

```
CALCULATED TOP OF SURFACE TRANSPORT LAYER IS:     1106.12
USE A SIGMA FOR WIND AZIMUTH ANGLE OF:     30.0          YES       NO

USE A SIGMA FOR WIND AZIMUTH ANGLE OF:     30.0          YES

CENTERLINE CONCENTRATION-PLOT DESIRED ?                  YES       NO

YES
  CENTERLINE CONCENTRATION PLOT DESIRED ?                YES
```

Figure 3-11. Centerline Concentration/Dosage Plots Added

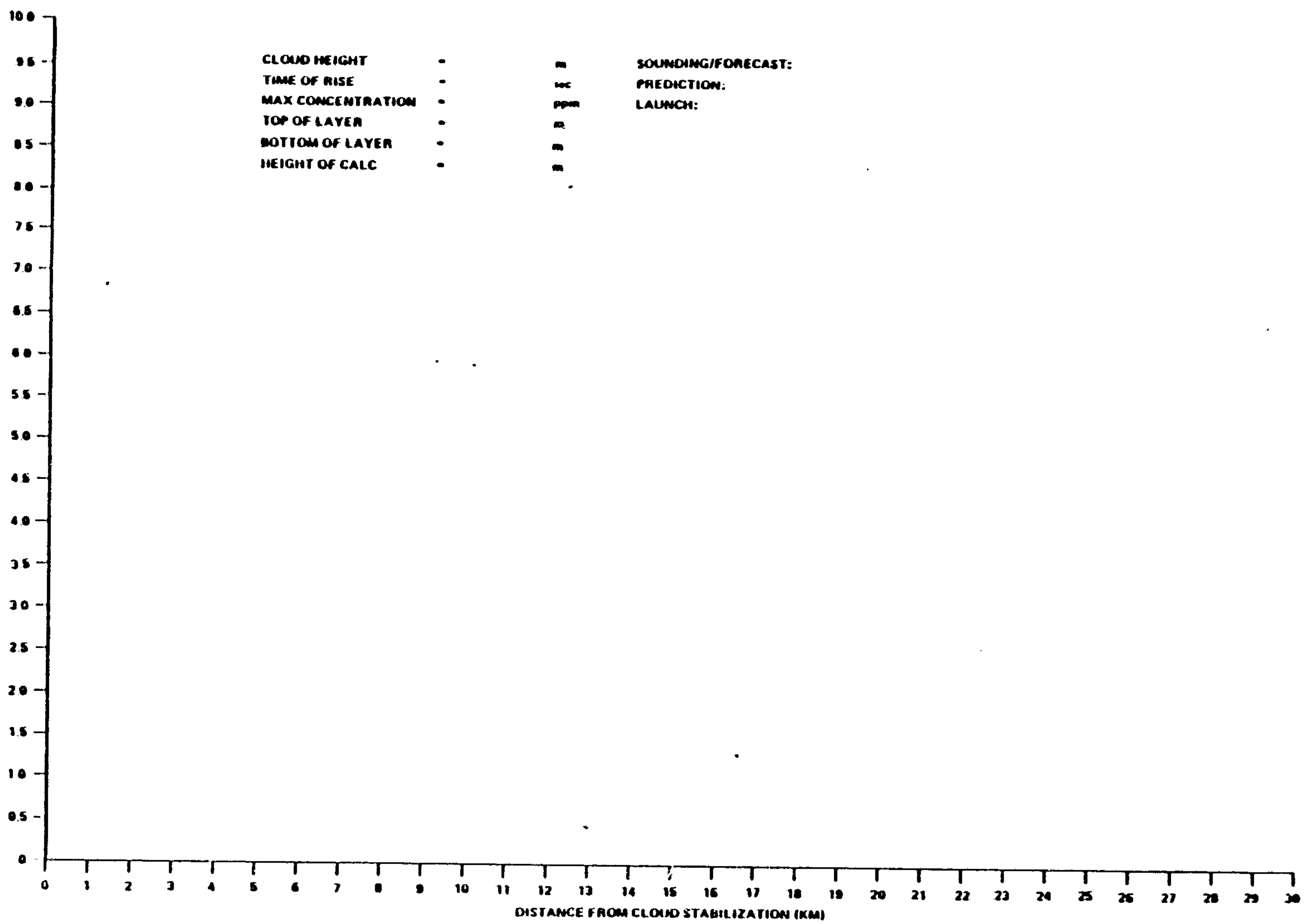


Figure 3-12. Blank Form for Centerline Concentration/Dosage Plots

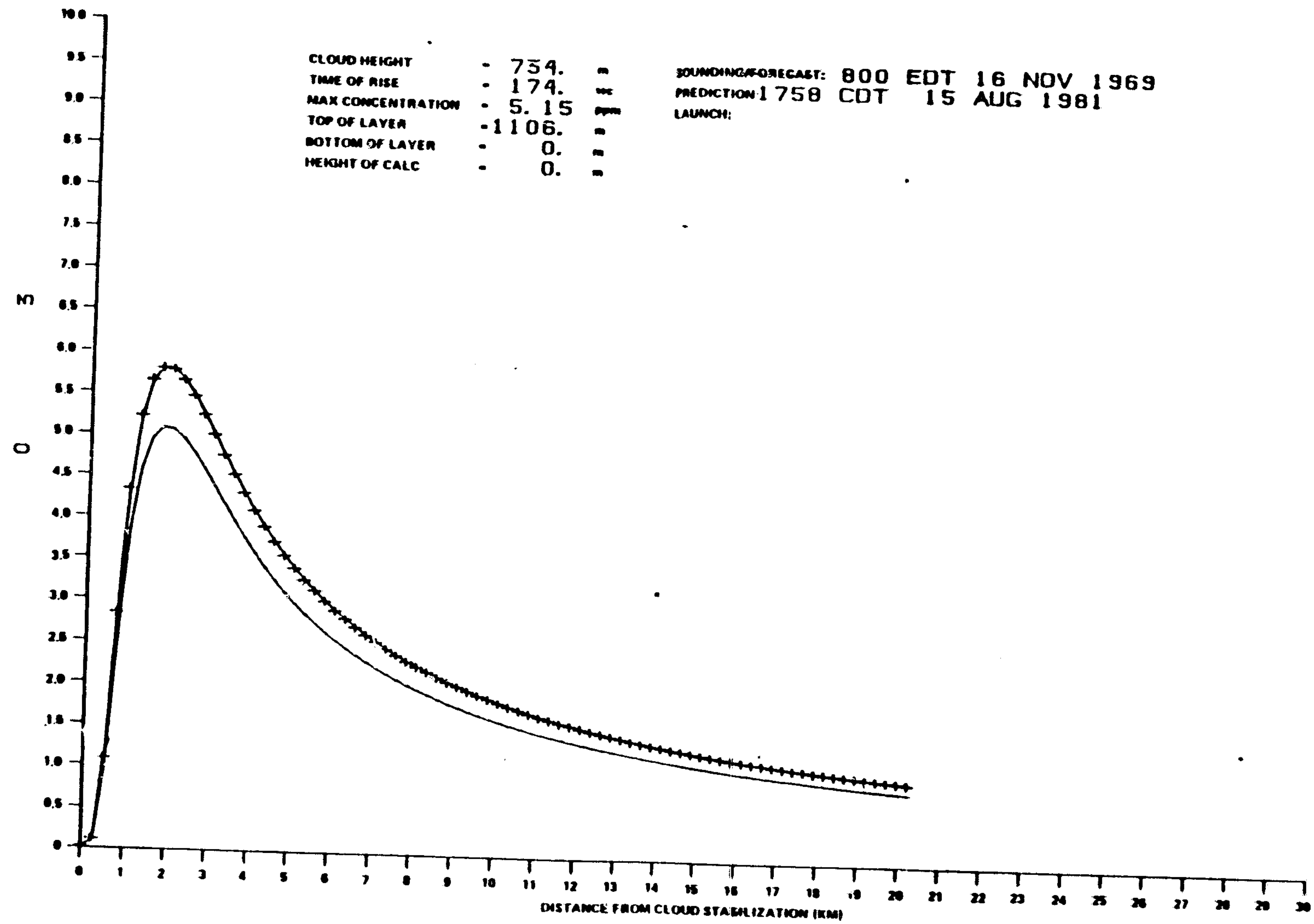


Figure 3-13. Centerline Dosage/Concentration Plots

CLOUD CONCENTRATIONS AND DOSAGES

DISTANCE (METERS)	CONCENTRATION (PPM)	DOSAGE (PPM SEC)	TIME AFTER LAUNCH(SEC) START	FINISH	POSITION FROM PAD RANGE	AZ
0.0	.000	.057	-1787	2135.	111.7	195.3
1000.0	3.831	4379.79	-261.	3661.	1111.4	191.7
2000.0	5.126	5859.96	1264.	5186.	2111.4	191.6
3000.0	4.432	5066.42	2790.	6712.	3111.4	191.5
4000.0	3.618	4135.67	4316.	8238.	4111.4	191.5
5000.0	3.005	3435.43	5841.	9763.	5111.4	191.4
6000.0	2.564	2931.11	7367.	11289	6111.4	191.4
7000.0	2.235	2555.21	8893.	12815	7111.4	191.4
8000.0	1.981	2264.59	10418	14340	8111.4	191.4
9000.0	1.779	2033.24	11944	15866	9111.4	191.4
10000.0	1.614	1844.72	13470	17392	10111.4	191.4
11000.0	1.477	1688.16	14995	18917	11111.4	191.4
12000.0	1.361	1556.07	16521	20443	12111.4	191.4
13000.0	1.262	1443.14	18047	21969	13111.4	191.4
14000.0	1.177	1345.48	19572	23495	14111.4	191.4
15000.0	1.102	1260.19	21098	25020	15111.4	191.4
16000.0	1.037	1185.07	22624	26546	16111.4	191.4
17000.0	.978	1118.39	24150	28072	17111.4	191.4
18000.0	.926	1058.82	25675	29597	18111.4	191.4
19000.0	.879	1005.27	27201	31123	19111.4	191.4
20000.0	.837	956.869	28727	32649	20111.4	191.4

****POINT OF MAXIMUM CONCENTRATION****
RANGE FROM PAD(M): 1861.4
DIRECTION(DEG): 191.6
HEIGHT(M): 0.0
MAXIMUM CONCENTRATION(PPM): 5.145

****POINT OF MAXIMUM DOSAGE****
RANGE FROM PAD(M): 1851.4
DIRECTION(DEG): 191.6
HEIGHT(M): 0.0
MAXIMUM DOSAGE (PPM SEC): 5882.0

****CONCENTRATIONS AND DOSAGES WITH 10 DEGREE UNCERTAINTIES****

RANGE(M): 5000.0
AZIMUTH(DEG): 170.0

MATERIAL	CONCENTRATION(PPM)			DOSAGE(PPM)		
HCL	.184E+01	+/-	.050	2109.108	+/-	57.684
CO	.676E-02	+/-	.000	7.730	+/-	.211
CO2	.403E+01	+/-	.110	4606.357	+/-	125.984
AL2O3	.473E+01	+/-	.129	5409.617	+/-	147.953
NO	.322E-02	+/-	.000	3.681	+/-	.101

RANGE(M): 5000.0
AZIMUTH(DEG): 190.0

MATERIAL	CONCENTRATION(PPM)			DOSAGE(PPM)		
HCL	.306E+01	+/-	.285	3494.906	+/-	326.191
CO	.112E-01	+/-	.001	12.809	+/-	1.195

Figure 3-14. Addition of Maximum Points of Concentrations and Dosage Along Ground Track

Off-Centerline Concentration/Dosages. The user is then prompted for off-centerline concentrations/dosages. These are selected, causing a call to the ORIGIN subroutine and a further prompt for the desired pad number/map type. Of the choices listed, pad 39 was chosen, and a sea map (Figure 3-15) rather than a land map (Figure 3-16 is chosen. As will be seen, this is a mistake, as better resolution and coverage would result for a land map.

The user is then prompted for range and azimuth (pad-relative) to the point for which off-centerline calculations are desired. The prompting continues (Figure 3-17) until a zero (0) range is entered. During this time, the off-centerline effluent concentrations, their associated $\pm 10^{0}$ (in azimuth) uncertainties, and dosages and their respective uncertainties, are printed on the printer (Figure 3-18). The points are also plotted on the plotter, and labeled with their ordinate index (1st one = 1, 2nd = 2, etc.) (Figure 3.19).

Isopleth Plotting. The user is then prompted for isopleth plotting. This is accepted and the default set of concentration levels is listed on the terminal display, and the user is asked if he intends to use these, or input his own (Figure 3-20). These are accepted, and the isopleths are drawn and (Figure 3-21) labeled.

Cloud Arrival/Departure Time. Finally, the cloud position and extent as a function of time are printed in a concluding table (Figure 3-22).

Termination. The user is then asked for termination ("T") or restart ("R") termination is chosen (Figure 3-20).

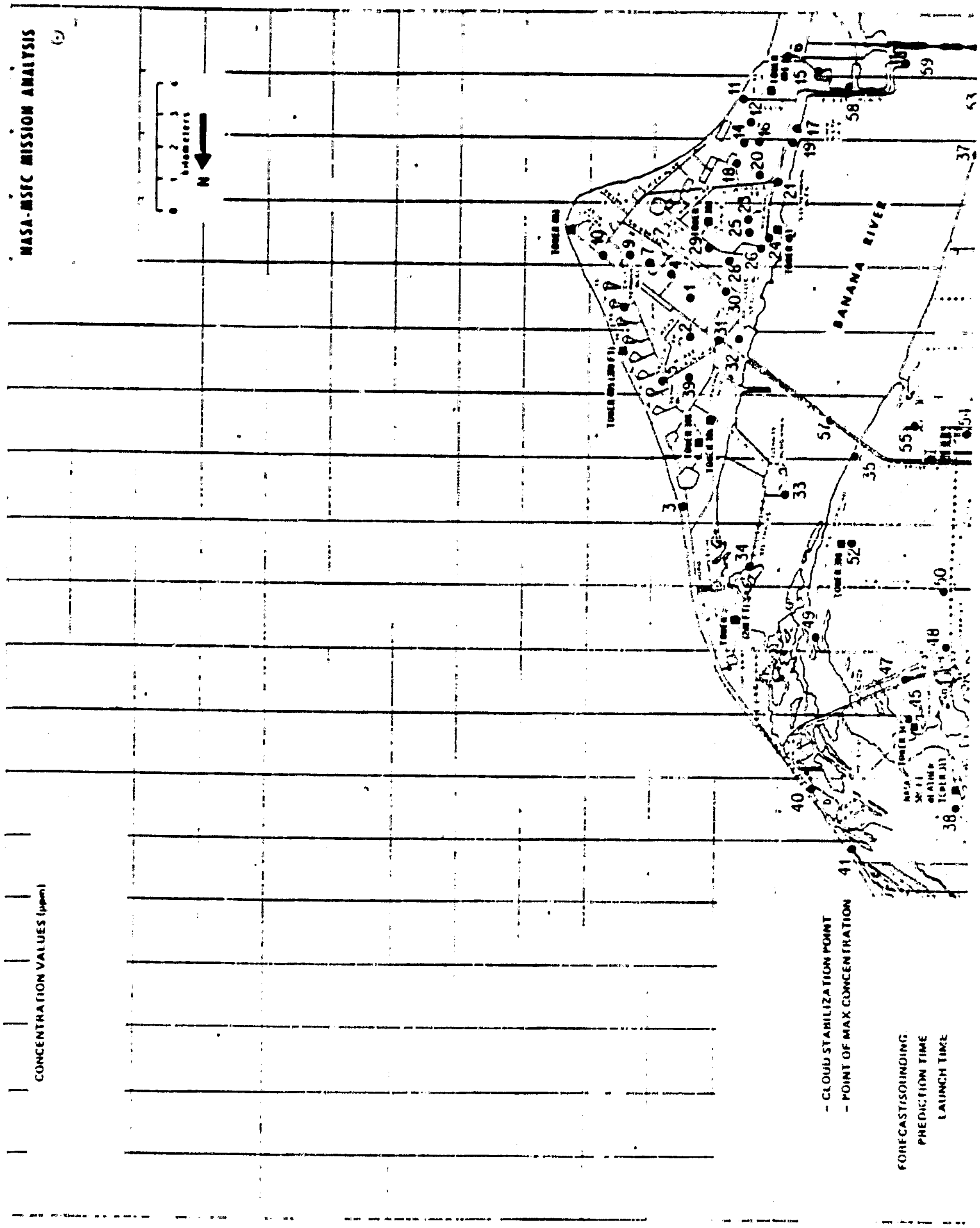


Figure 3-15. Blank Sea Map

ORIGINAL PAGE IS
OF POOR QUALITY

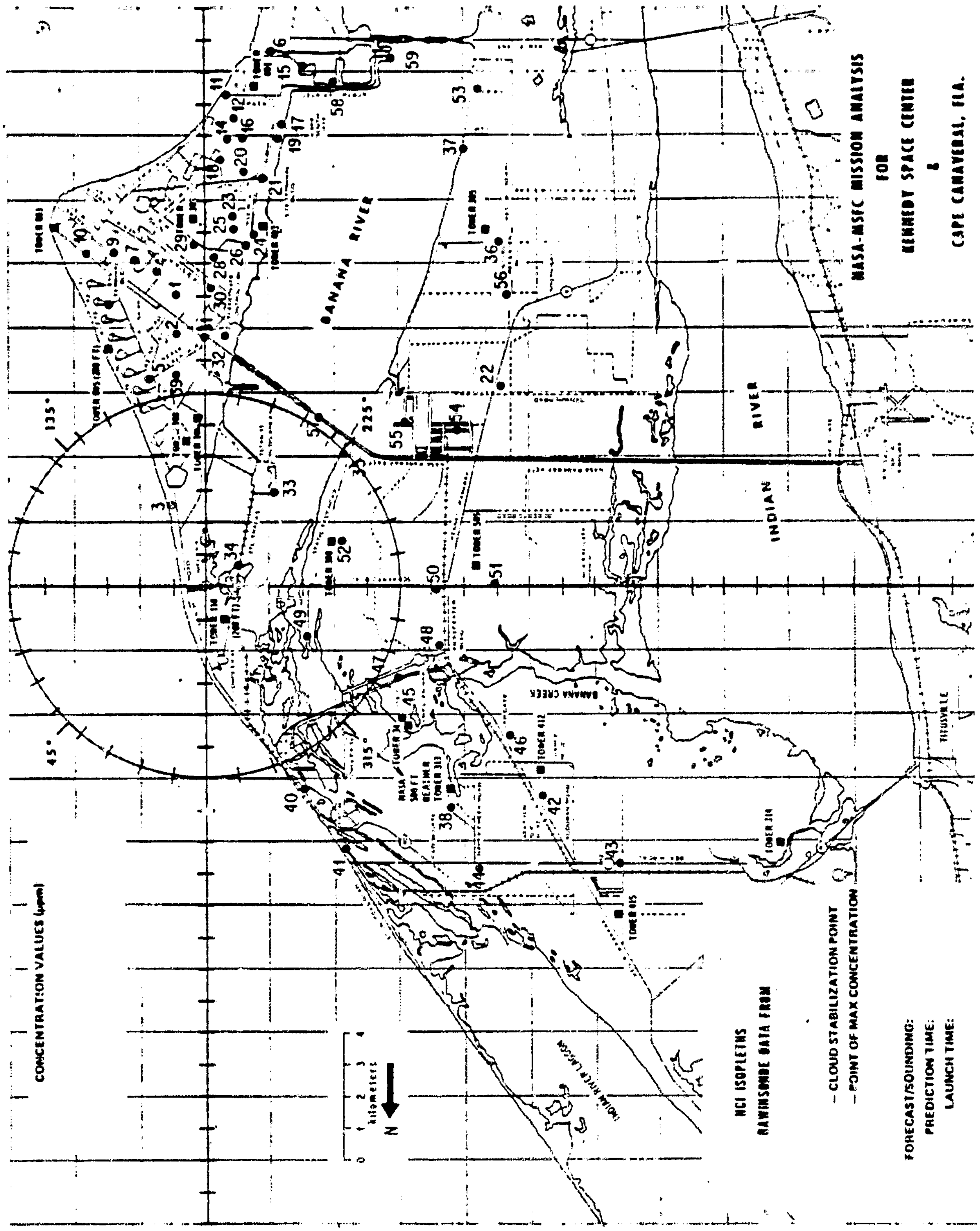


Figure 3-16. Blank Land Map

```
SPECIFIC SITE PREDICTIONS (PLOTTED)?                                   YES      NO


                                   39.SEA MAP

ENTER RANGE(M) AND AZIMUTH(DEG):  (Ø TERMINATES PROCEDURE)
     RANGE:  1000            AZIMUTH:  192
     RANGE:                  AZIMUTH


ENTER RANGE(M) AND AZIMUTH(DEG):  (Ø TERMINATES PROCEDURE)
     RANGE:  1000            AZIMUTH:  192
     RANGE:  2500            AZIMUTH:  192
     RANGE:  5000            AZIMUTH:  192
     RANGE:  7000            AZIMUTH:  192
     RANGE:  8500            AZIMUTH:  192
     RANGE:  10000           AZIMUTH:  192
     RANGE:  12000           AZIMUTH:  192
     RANGE:  15000           AZIMUTH:  192
     RANGE:  20000           AZIMUTH:  192
     RANGE:  25000           AZIMUTH:  192
     RANGE:  0               AZIMUTH:

     RANGE:  0               AZIMUTH:

CONCENTRATION ISOPLETH PLOT DESIRED?                                   YES      NO
```

Figure 3-17. Specific Site Range/Azimuth Entry

****CONCENTRATIONS AND DOSAGES WITH 10 DEGREE UNCERTAINTIES****

RANGE(M): 5000.0
AZIMUTH(DEG): 170.0

MATERIAL	CONCENTRATION(PPM)		DOSAGE (PPM)	
HCL	.184E+01 +/-	.050	109.108 +/-	57.684
CO	.676E-02 +/-	.000	7.730 +/-	.211
CO2	.403E+01 +/-	.110	4606.357 +/-	125.984
AL2O3	.473E+01 +/-	.129	5409.617 +/-	147.953
NO	.322E-02 +/-	.000	3.681 +/-	.101

RANGE(M): 5000.0
AZIMUTH(DEG): 190.0

MATERIAL	CONCENTRATION(PPM)		DOSAGE (PPM)	
HCL	.306E+01 +/-	.285	3494.906 +/-	326.191
CO	.112E-01 +/-	.001	12,809 +/-	1.195
CO2	.668E+01 +/-	.623	7632.983 +/-	712.411
AL2O3	.784E+01 +/-	.732	8964.029 +/-	836.641
NO	.534E-02 +/-	.000	6.099 +/-	.569

RANGE(M): 7500.0
AZIMUTH(DEG): 190.0

MATERIAL	CONCENTRATION(PPM)		DOSAGE (PPM)	
HCL	.212E+01 +/-	.219	2428.497 +/-	250.218
CO	.779E-02 +/-	.001	8.900 +/-	.917
CO2	.464E+01 +/-	.478	5303.912 +/-	546.484
AL2O3	.545E+01 +/-	.561	6228.813 +/-	641.781
NO	.371E-02 +/-	.000	4.238 +/-	.437

RANGE(M): 10000.0
AZIMUTH(DEG): 190.0

MATERIAL	CONCENTRATION(PPM)		DOSAGE (PPM)	
HCL	.163E+01 +/-	.176	1859.782 +/-	201.469
CO	.596E-02 +/-	.001	6.816 +/-	.738
CO2	.355E+01 +/-	.385	4061.821 +/-	440.015
AL2O3	.417E+01 +/-	.452	4770.125 +/-	516.745
NO	.284E-02 +/-+/-	.000	3.246+/-	.352

RANGE(M): 12500.0
AZIMUTH(DEG): 190.0

MATERIAL	CONCENTRATION(PPM)		DOSAGE (PPM)	
HCL	.132E+01 +/-	.147	1506.710 +/-	168.214
CO	.483E-02 +/-	.001	5.522 +/-	.616
CO2	.288E+01 +/-	.321	3290.701 +/-	367.386
AL2O3	.338E+01 +/-	.377	3864.536 +/-	431.451
NO	.230E-02 +/-+/-	.000	2.630 +/-	.294

Figure 3-18. Off-Centerline Concentrations and Dosages with $\pm 10^{\circ}$ Uncertainties for all Effluents

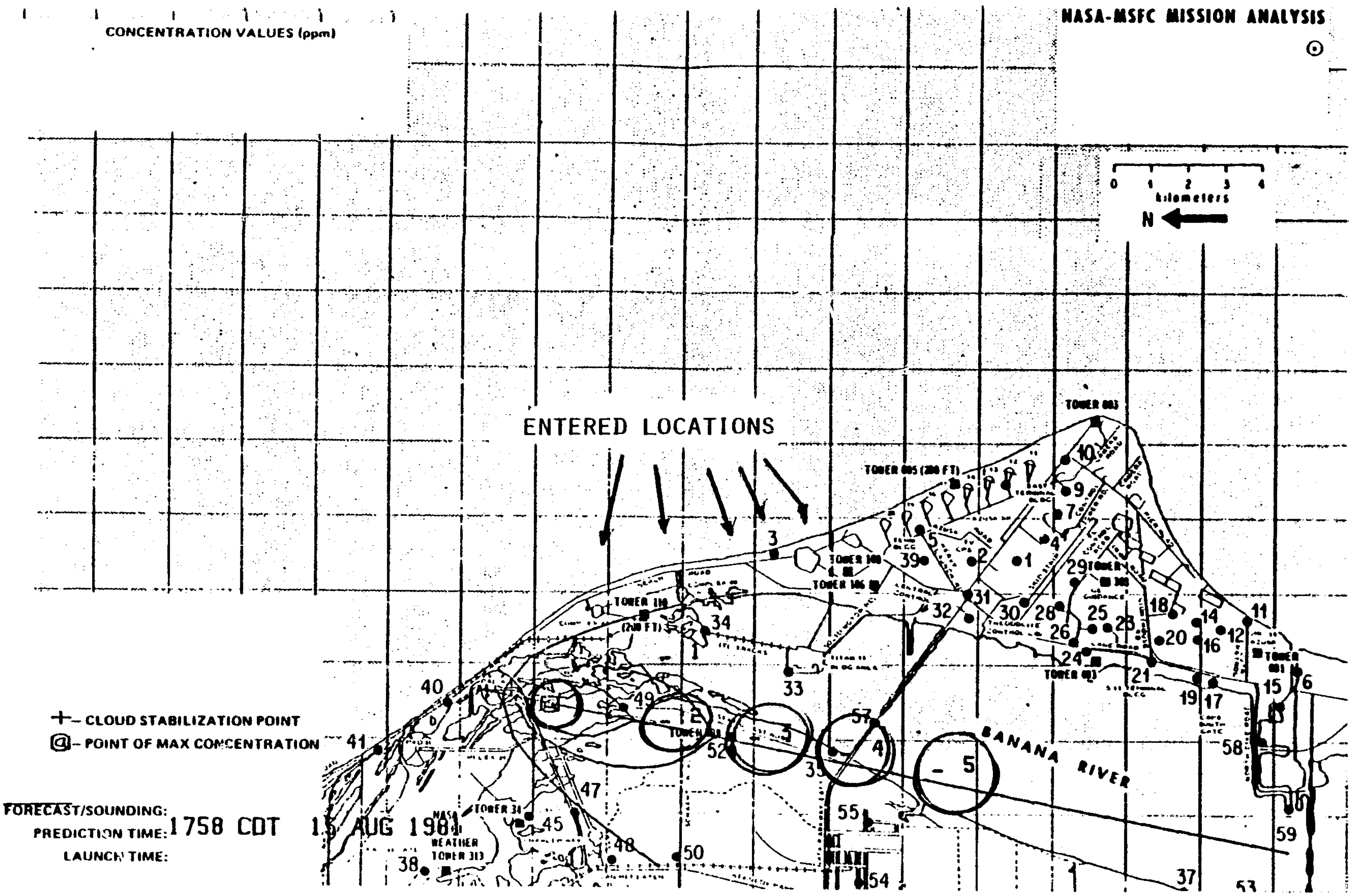


Figure 3-19. Off-Centerline Point Plotting

```
CONCENTRATION ISOPLETH PLOT DESIRED?                        YES       NO


YES


 DEFAULT ISOPLETH CONCENTRATION VALUES ARE:  CONC(1) = .515
     CONC(2)=2.573     CONC(3)=3.859     CONC(4)=1.0

 USE DEFAULT ISOPLETH CONCENTRATION VALUES ?                YES       NO


YES

T

 *****  PROGRAM REEDS HAS TERMINATED NORMALLY ****
 REE05 : STOP 0000
```

Figure 3-20. Default Isopleth Values, Isopleth Plotting, and Program Termination

ORIGINAL PAGE IS
OF POOR QUALITY

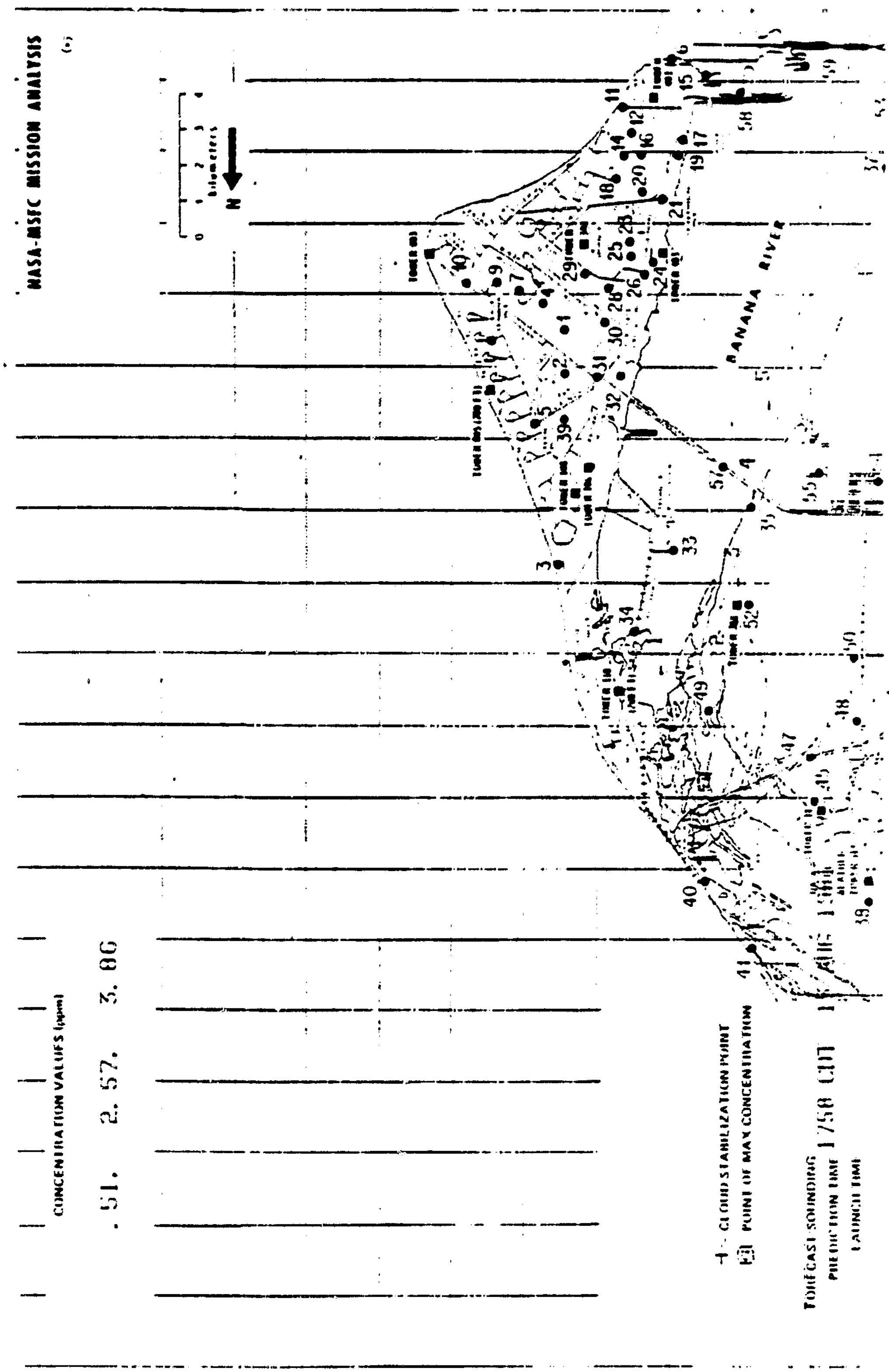


Figure 3-21. Isopleth Plotting

CLOUD LOCATION AND DIMENSIONS

TIME FROM LAUNCH (MIN)	(SEC)	RANGE (METERS)	AZIMUTH (DEG)	DIAMETERS (METERS) CROSS WIND	ALONG WIND
THE DISTANCE FROM STABILIZATION IS:		0.000			
2	54.0	119.8	201.3	1285.3	1285.3
THE DISTANCE FROM STABILIZATION IS:		196.63			
7	54.0	304.4	180.9	1515.0	1285.6
THE DISTANCE FROM STABILIZATION IS:		393.27			
12	54.0	498.3	176.1	1756.7	1286.4
THE DISTANCE FROM STABILIZATION IS:		589.90			
17	54.0	693.8	174.0	2006.1	1287.7
THE DISTANCE FROM STABILIZATION IS:		786.53			
22	54.0	889.8	172.8	2260.8	1289.5
THE DISTANCE FROM STABILIZATION IS:		983.17			
27	54.0	1086.0	172.1	2519.0	1291.9
THE DISTANCE FROM STABILIZATION IS:		1179.80			
32	54.0	1282.3	171.5	2779.8	1294.7
THE DISTANCE FROM STABILIZATION IS:		1376.44			
37	54.0	1478.7	171.2	3042.6	1298.1
THE DISTANCE FROM STABILIZATION IS:		1573.07			
42	54.0	1675.2	170.9	3306.9	1302.0
THE DISTANCE FROM STABILIZATION IS:		1769.70			
47	54.0	1871.7	170.6	3572.3	1306.4
THE DISTANCE FROM STABILIZATION IS:		1966.34			
52	54.0	2068.2	170.4	3838.6	1311.3
THE DISTANCE FROM STABILIZATION IS:		2162.97			
57	54.0	2264.8	170.3	4105.6	1316.7
THE DISTANCE FROM STABILIZATION IS:		2359.60			
62	54.0	2461.3	170.2	4373.2	1322.5

Figure 3-22. Cloud Position and Extent as a Function of Time

SECTION 4.0 LISTINGS

This section contains the current listings of critical elements of the REEDS system. They are included as follows:

SECTION 4.1 Main Program Modules and Subroutines Packages

- 4.1.1 REEDS (Executive Module)
- 4.1.2 REEDT
- 4.1.3 RCLDS
- 4.1.4 RCLDT
- 4.1.5 RMTPS
- 4.1.6 RMTPT
- 4.1.7 RCONS
- 4.1.8 RCONT
- 4.1.9 RISOS
- 4.1.10 RISOT

SECTION 4.2 AUXILIARY SUBROUTINE LISTINGS

- 4.2.1 CTLB2
- 4.2.2 MLIB2
- 4.2.3 CALIX
- 4.2.4 DLINE

SECTION 4.3 MAIN MODULE QUESTION FILES

- 4.3.1 ?REEDS
- 4.3.2 ?RCLDS
- 4.3.3 ?RCONS
- 4.3.4 ?RISOS

SECTION 4.4 EXAMPLE METEOROLOGICAL DATA FILES

4.1 MAIN PROGRAM MODULES AND SUBROUTINE PACKAGES

This section contains the current listings of current Main Program Modules and the respective Main Subroutine Packages. They are listed according to the sequences indicated in Section 4.0.

```
&REEDS T=00003 IS ON CR00003 USING 00150 BLKS R=0000

FTN4X,L
      PROGRAM REEDS(3),REEDS SINGLE LAYER DIFFUSION MODEL--CA881 0825
C
C ************************************************************
C *                                                          *
C *    NASA/MSFC REED Code --             -- EXPERIMENTAL    *
C *                                                          *
C *         SINGLE LAYER DISTRIBUTION TECHNIQUE              *
C *                                                          *
C *    MAIN PROGRAM -- REEDS                                 *
C *                                                          *
C ************************************************************
C
C ************************************************************
C *                                                          *
C * LOADING INSTRUCTIONS:                                    *
C *                                                          *
C *  REEDS, RCLDS, RCONS, AND RISOS ARE LOADED AS FOLLOWS:   *
C *    LOAD %RXXXS                                           *
C *    MERGE %RXXXT                                          *
C *    MERGE %LINE                                           *
C *    MERGE %DLINE                                          *
C *    MERGE %CALIX                                          *
C *    MERGE %CTLB2                                          *
C *    END                                                   *
C *                                                          *
C *  RMTPS IS LOADED AS FOLLOWS:                             *
C *    LOAD %RMTPS                                           *
C *    MERGE %RMTPT                                          *
C *    MERGE %DLINE                                          *
C *    MERGE %CALIX                                          *
C *    MERGE %MLIB2                                          *
C *    END                                                   *
C *                                                          *
C ************************************************************
C
C >>>   COMMON AREAS
C
C
C >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGMEL
C
C >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNGFG,IVHICL,IRUN,ICALC,IPLACE,ITIME2,IGOTNP,
     .       NUMRUN,LU,NUM,MONLY,MFORM,LUPPNT,ITDU,IFL1,IFL2,IFL3
      INTEGER YES
C
C >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SUPDEN,FILE(3)
      INTEGER FILE
C
C >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
C  >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C  >>>   LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME, LVERSN
C
C  >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIN(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C  >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C  >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C  >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PHO,VPAR(17)),(GAMMAX,VPAR(18))
      DIMENSION ILINE(32),IDATAF(10),IERS(32),JJTIN(5),IMONG(5,2)
      DIMENSION IDAYG(5),ITIMG(5)
C
C        INPUT FORMAT STATEMENTS
C
90    FORMAT(A1)
91    FORMAT(A2)
  100 FORMAT (I2,1X,2A2,I2)
  101 FORMAT (A1)
  102 FORMAT (10A2)
  103 FORMAT (I4,5XI2,1XA2,A1,1XI4)
  104 FORMAT (I4,3XI2,1XA2,A1,1XI4)
  105 FORMAT (I6,1XI3,1XF4.1,F6.1,F6.1,F7.2,11XF7.2)
  106 FORMAT (I4," C"A2,1XI2,1X,A2,A1,1X,I4)
  107 FORMAT (I2,1X,A2,A1,1X,I4)
  108 FORMAT (F7.2)
  109 FORMAT (I2)
  110 FORMAT (F4.1)
  111 FORMAT (I2,1X,A2,A1,1X,I4)
C
C        OUTPUT FORMAT STATEMENTS
C
  200 FORMAT ("1"80(1H*)/1X,80(1H*)/1X,10(1H*),60X,10(1H*)/
     .        1X,10(1H*)," NASA/MSFC REED CODE USING SINGLE LAYER "
     .        "MODEL"3X"21 DEC 1979 ",10(1H*)/1X,10(1H*),60X,10(1H*)/
     .        1X,10(1H*),60X,10(1H*)/1X,10(1H*),27X"REEDS "27X,10(1H*)/
     .        1X,10(1H*),60X,10(1H*)/1X,80(1H*)/1X,80(1H*)/
     .        "0PUN "I2" USING DATA FILE "3A2/
     .        "0"3A2,A1" LAUNCH VEHICLE")
  201 FORMAT ("0LAUNCH TIME: "I4,1XR1,A2,4X"DATE: "I2,1XA2,A1,1XI4)
  202 FORMAT ("0PREDICTION TIME: "I4" C"A2,4X"DATE: "I2,1XA2,A1,1XI4/
     .        "0DATA FILE HEADER INFORMATION:")
  203 FORMAT ("0<<<<OPEN ERROR "I4", PROCESSING CONTINUES WITH "
     .        "NEXT RUN>>>>")
  204 FORMAT ("0<<<<READ ERROR "I4", PROCESSING CONTINUES WITH "
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      .           "NEXT RUN>>>")
  205 FORMAT (6X,40A2)
  206 FORMAT ("0"5X"TIME: "I4,1XA1,A2,4X"DATE: "I2,1XA2,A1,1XI4)
  207 FORMAT (//"0"80(1HS)/1X,20(1HS),40X,20(1HS)/
      .           1X,20(1HS),16X"SOUNDING"16X,20(1HS)/
      .           1X,20(1HS),40X,20(1HS)/1X,80(1HS)//)
  208 FORMAT (//"0"80(1HF)/1X,20(1HF),40X,20(1HF)/
      .           1X,20(1HF),16X"FORECAST"16X,20(1HF)/
      .           1X,20(1HF),40X,20(1HF)/1X,80(1HF)//)
  209 FORMAT ("0SURFACE DENSITY (GM/M**3): "F8.2)
  210 FORMAT ("0LAYER   ALTITUDE     DIRECTION  SPEED   TEMP   "
      .           "POT-TEMP  D P TEMP  PRESSURE"/
      .           "  NO. (FEET) (METERS) (DEGREES) (M/SEC)    "
      .           "(DEGREES CENTIGRADE)    (MILLIBARS)")
  211 FORMAT (2XI2,I7,2XI5,7XI3,5XF4.1,4XF4.1,4XF5.2,6XF4.1,6XF5.2)
  212 FORMAT ("****OPEN OR IDRPL IERR: "I3", JOB TERMINATED****")
  213 FORMAT ("****IDRPD IERR: "I3", JOB TERMINATED****")
C
C          TYPE AND DIMENSION STATEMENTS
C
      INTEGER BLANKS,RCHAR,TCHAR,SCHAR,VNAMES(4,3),RUNNUM,RA,
     .        FO,SDT,TE,ZERO6,ZERO1,RCLDR(3),NAMRP(3,5)
      DIMENSION IPAR(5),VPARS(18,5),IDCB(272),IBUF(40),IALT(30),
     .          DPTEMP(30),NAME(3),NAMEF(3)
C
C          DATA STATEMENTS
C
      DATA NAME/2H=R,2HEE,2HD2/
      DATA IERS/32*2H  /
      DATA NAMEF/2H?R,2HEE,2HDS/
C
C****************************************************************************
C***  VPARS(1 THRU 18) = SHUTTLE    (19 - 36) = TITAN    (37 - 54)=DELTA
C***       (55 - 72) = DELTA 3914    (73 - 90) = MINUTEMAN
C****************************************************************************
C
C     VALUES CONSISTENT WITH SAI REVISION OF SS EIS---2/22/78
C     QC1,QT1 ARE VALUES FOR NORMAL LAUNCH SEQUENCE
C     QC2,QT2 ARE VALUES FOR ONE SRM BURNING ON LAUNCH PAD
C     QC3,QT3 ARE VALUES FOR BOTH SRMS EXPLODING ON LAUNCH PAD
C     VALUES ARE CONSISTENT WITH THIOKOL VAB SAFETY STUDY
C     BURN TIME IS 7.7778 TIMES NORMAL BURN TIME
C     CC IS HOLD DOWN TIME ON LAUNCH PAD FOR NORMAL LAUNCH
CC    DATA VPARS/9.47059090E6,3.0450568E6,9.0072607I1E5,1.251174E9,
CC   .           5.075475E8,1.015095E9,.652213,.468085,
CC   .           .29653,1479.7,1062.35,1000.,.1146,.00042,.25029,.18279,
CC   .           .00319,.64,
CCC   DATA VPARS/1.5454E7,6.082960E6,9.0072607I1E5,1.894794173E9,
      DATA VPARS/1.5219E7,6.082960E6,9.0072607I1E5,1.251174E9,
CCC  .           8.569295E9,1.713859E9,.652213,.468095,
     .           5.075475E8,1.015095E9,.652213,.468095,
CCC  .           .29653,1479.7,1062.35,1000.,.1146,.00042,.25029,.18279,
     .           .37500,1479.7,1062.35,1000.,.1146,.00042,.25029,.18279,
CCC  .           .00819,.64,
     .           .00020,.64,
     .           5.437520E6,2.718764E6,1.35[illegible]E6,3.[illegible]E0,
     .           1.6312594E8,3.26[illegible]E8,.4[illegible],.51[illegible],
CCC  .           5.0,2021.1,1010.5[illegible],1000.0,.1[illegible],.[illegible],.0[illegible],
     .           5.0,2021.1,1010.55,1000.0,.1[illegible],.0[illegible],.[illegible],
     .           [illegible]
```

```
                  8.360685E5,9.09811E4,2.729434E5,2.887538E7,
                  3.14229E6,1.885373E7,.922156,.432703,.54,1766.0,
      CCC   .     1000.0,690.0,.1866,.2055,.0156,.3391,.0002,
            .     1000.0,690.0,.1210,.0156,.2055,.2214,.0002,
            .     .50,
            .     1.057557E6,1.482923E5,3.70731E5,6.70269E7,
            .     9.398616E6,4.699308E7,1.245756,.4180947,
      CCC   .     0.0,1449.9,1000.0,411.10,.1866,.2055,.0156,.3391,
            .     0.0,1449.9,1000.0,411.10,.1589,.0331,.2703,.1936,
            .     .0002,.50,
            .     4.684476E5,4.684476E5,1.171119E5,2.8106856E7,
            .     2.8106856E7,2.8106856E7,.469982,.463333,0.0,
      CCC   .     2055.9,2055.9,1000.0,.1866,.2055,.0156,.3391,
            .     2055.9,2055.9,1000.0,.1866,.0156,.2055,.3391,
            .     .0002,.64/
      C
            DATA BLANKS/2H  /, RCHAR/1HR/, TCHAR/1HT/, SCHAR/1HS/,
            .    RA/2HRA/, FO/2HFO/, SDT/2HDT/, TE/2HTE/, ZERO0/30060B/,
            .    ZERO1/034471B/, RCLDR/2HRC,2HLD,1HS/
            DATA VNAMES/2HSH,2HUT,2HTL,1HE,
            .         2HTI,2HTA,1HN,1H ,
            .         2HDE,2HLT,1HA,1H /
            DATA NAMRP/2HRC,2HLD,2HS ,2HRM,2HTP,2HS ,2HRC,2HON,2HS ,
            .         2HRI,2HSO,2HS ,2HRM,2HTQ,2HS /
      C
      C        SET THE LOGICAL UNIT NUMBERS OF THE DEVICES TO BE USED FOR
      C        INPUT AND PRINT

      C
            CALL RMPAR(IPAR)
            LU = MAX0(IPAR(1),1)
            LUPRNT = IPAR(2)
            IF(LUPRNT .LE. 0)LUPRNT = 6
            IPL1 = IPAR(3)
            IF(IPL1.LE.0) IPL1 = 12
            IPL2 = IPAR(4)
            IF(IPL2.LE.0) IPL2 = 20
            IPL3 = IPAR(5)
            IF(IPL3.LE.0) IPL3 = 21
      C
      C        FIND OUT IF THIS IS THE CRT OR PLASMASCOPE VERSION OF REEDS
      C        BEING RUN
      C
            CALL VERSN(I)
            LVERSN = I .EQ. 0
      C
      C        RP ALL SEGMENTS -- IF ERRORS ENCOUNTERED, WRITE OUT AN ERROR
      C        MESSAGE AND TERMINATE REEDS
      C
            NUMSEG = 4
            IF(LVERSN)NUMSEG = 5
            DO 1 I=1,NUMSEG
            CALL OPEN(IDCB,IERR,NAMRP(1,I),1)
            CALL IDRPL(IDCB,IERR,NAMRP(1,I))
            IF(IERR .GE. 0)GO TO 1
            WRITE (LU,212) IERR
            STOP
          1 CONTINUE
      C
      C        FOR PLASMASCOPE ONLY, CALL [illegible]
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C         GRAF FOR CRT
C
      CALL GRAF(1)
C
C         BEGIN PROCESSING OF NEW DATA BY CLEARING SCOPE
C
    2 CALL CLEAR
C
C         INITIALIZE SOME COMMON VARIABLES
C
      LTIME = .FALSE.
      YES = 1HY
      NO = 1HN
      PI = 3.141593
      PIOVR2 = 0.5 * PI
      PI43 = 1.3333333 * PI
      TWOPI = 2.0 * PI
      SQR2PI = SQRT(TWOPI)
      DEGRAD = PI/180.0
      RADDEG = 180.0/PI
      ISNDFO = 0
      IVHICL = 0
      IRUN = 0
      ICALC = 0
      IPLACE = 0
      ITIMEZ = 0
      IGOTMP = 0
      LAYBOT = 0
      BOTLAY = 0.0
      CALHT = 0.0
CBJ   REFLEC = 1.0
      REFLEC = 0.1
      IPTZ = 0
C
C         WRITE THE HEADER ON THE CONSOLE
C
      CALL CLEAR
      YPOS = 490.0
      CALL DREAD(NAMEF,2,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
      CALL GETTD(LTIM,LDAY,LMONTH,LYEAR)
      CALL CODE
      WRITE (IDATAF,107) LDAY,LMONTH,LYEAR
      CALL CHAR(368.0,YPOS,0,IDATAF,11,2,0)
      YPOS = YPOS - 32.0
CBJ
CBJ*** DECIDE ON THE TYPE OF CLOUD RISE DESIRED...NEW CURVE FIT OR
CBJ*** OLD PROCEDURE
      WRITE(1,777)
777   FORMAT(" DO YOU WANT THE LOW ALT  CURVE FIT? YES  NO _")
      READ(1,776) IANS
776   FORMAT(A1,"_")
      IPTZ = 1
      IX = 25
      IF(IANS .NE. 1HY) GO TO 771
      IX = 30
      IPTZ = 0
      WRITE(1,774)
774   FORMAT(" LOW ALTITUDE CURVE FIT ...")
```

```
      GO TO 770
771   CONTINUE
      WRITE(1,773)
773   FORMAT(" LOW ALTITUDE CURVE F37",30X," NO")
770   CONTINUE
C
      YPOS = YPOS - 32.0
C        READ IN THE 1ST 4 CHAR'S OF THE DATA FILE NAMES
C
      CALL DREAD(NAMEF,3,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,43,3,0)
      CALL CHAR(384.0,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.0,YPOS,0,ILINE(30),6,0,0)
C
C  IF PLASMASCOPE VERSION, CALL "IN".
C
      IF(LVERSN)GO TO 3
C
C  OTHERWISE RESPOND WITH Y FOR YES, N FOR NO TO ANSWER QUESTIONS
C***  A  "0"  FOR ALMOST ANY ENTRY WILL PERMIT IMMEDIATE TERMINATION
C***  OF THE PROGRAM
C***
      READ(LU,90) IANS
      IF(IANS.EQ.1H0)GO TO 81
C
C  "YES" IS THE DEFAULT CASE
C
      IX = 25
      IF(IANS .NE. NO) GO TO 4

C  THIS IS THE "NO" RESPONSE---ENTER THE # OF RUNS, AND THE
C  COMMON DATA FILE NAME
C
      IX = 30
      GO TO 4
    3 CALL IN(1,JTYPE,0.0,0.0,0,0,0.0,31,0.31,IX,IY)
    4 CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)CALL CHAR(464.0,YPOS,0,IERS,6,0,0)
      IF(IX .GT. 25)CALL CHAR(384.0,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.0
      IF(IX .LE. 25)GO TO 6
      CALL DREAD(NAMEF,4,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 16.0
      CALL DREAD(NAMEF,5,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,22,3,0)
      NIN  = 9
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,175.0,YPOS,0,IDATAF,NIN,0.31,0.31,IX,IY)
      CALL CHAR(0.0,YPOS,0,ILINE,22,0,0)
      YPOS = YPOS - 32.0
      CALL CODE(NIN)
      READ (IDATAF,100) NUMRUN,FILE(1),FILE(2),IFOFF
      NUMRUN = MAX0(NUMRUN,1)
      IF(IFOFF .GT. 0)IFOFF = IFOFF - 1
      IF(FILE(1) .NE. BLANKS)GO TO 7
```

```
      FILE(2) = 2HTA
      IFOFF = 0
      NUMRUN = 1
    7 IFOFF1 = IFOFF + 1
      IPLACE = 0
      ISNDG = 0
CBJ
CBJ*** DECODE LOCATION FROM DATA FILE NAME
CBJ
      IF(FILE(1).EQ.2HVA .AND. FILE(2).EQ.2HND)IPLACE = 1
      IF(FILE(1).EQ.2HTA .AND. FILE(2).EQ.2HPE)IPLACE = 2
      IF(FILE(1).EQ.2HDA .AND. FILE(2).EQ.2HTA)IPLACE = 3
      IF(FILE(1).EQ.2HSP .AND. FILE(2).EQ.2HEC)IPLACE = 4
CBJ
CBJ*** ACCORDING TO PLACE, TIME ZONE CODE WILL BE ENCODED
      ITIMEZ = 1HE
      IF(IPLACE .EQ. 1)ITIMEZ = 1HP
C     IF(IPLACE .EQ. 2)REWIND 8
      IF(IPLACE.EQ.4) ISNDG = 1
      IF(IPLACE.NE.4)GO TO 9
C
CBJ*** IF THIS IS A SPECIFIC SOUNDING RUN, SOUNDING CODE WILL BE SET=1
C
C** ENTER SPECIFIC SOUNDING TIME AND DATE
C
      IF(NUMRUN.GT.5) NUMRUN = 5
      CALL DREAD(NAMEF,26,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.,YPOS,0,ILINE,46,3,0)
      DO 8 NSPEC=1,NUMRUN
      YPOS = YPOS - 16.
      NRUN = 26 + NSPEC
      CALL DREAD(NAMEF,NRUN,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.,YPOS,0,ILINE,20,0,0)
      CALL BLANK(IDATAF,10)
      NIN = 12
      CALL IN(0,JTYPE,160.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CODE(NIN)
    8 READ(IDATAF,111) IDAYG(NSPEC),IMONG(NSPEC,1),IMONG(NSPEC,2),
     .                 ITIMG(NSPEC)
      YMPOS = NUMRUN*16. + YPOS
      CALL DREAD(NAMEF,26,ILINE)
      CALL CHAR(0.,YMPOS,0,ILINE,46,0,0)
      YPOS = YPOS - 32.
C
C          FIND OUT IF THESE RUNS ARE TO BE RESEARCH RUNS (INTERACTION
C          AND PLOTTING ALLOWED) OR PRODUCTION RUNS
C
    9 CALL DREAD(NAMEF,7,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,12,3,0)
      CALL CHAR(96.0,YPOS,0,ILINE(7),18,0,0)
      CALL CHAR(240.0,YPOS,0,ILINE(16),14,3,0)
      CALL CHAR(352.0,YPOS,0,ILINE(23),18,0,0)
C
C*** CHECK FOR PLASMASCOPE.....
      IF(LVERSN)GO TO 10
C
C*** IF CRT, SET IY ACCORDING TO [illegible]
```

```
C*** OR P (PRODUCTION)
C
      IX = 15
      READ(LU,90) IANS
      IF(IANS.EQ.1H0)GO TO 81
      IF( (IANS.NE.1HR) .AND. (IANS.NE.1HP) ) GO TO 11
      IX = 10
      IF(IANS .NE. 1HP)GO TO 11
      IX = 20
      IF(IANS .EQ. 1HP)GO TO 11
 10   CONTINUE
      CALL IN(1,JTYPE,0.0,0.0,0,0,0,0,31,0,31,IX,IY)
 11   CONTINUE
      CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
      IF(IX .LT. 12)CALL CHAR(240.0,YPOS,0,IERS,32,0,0)
      IF(IX .LT. 12)IRUN = 1
      IF(IX.GE.12 .AND. IX.LT.19)CALL CHAR(120.0,YPOS,0,IERS,16,0,0)
      IF(IX.GE.12 .AND. IX.LT.19)IRUN = 2
      IF(IX.GE.12 .AND. IX.LT.19)CALL CHAR(368.0,YPOS,0,IERS,16,0,0)
      IF(IX .GE. 19)CALL CHAR(120.0,YPOS,0,IERS,30,0,0)
      IF(IX.GE.19) IRUN = 3
      YPOS = YPOS - 32.0
      IF(IRUN .LE. 2)GO TO 12
C
C        FOR PRODUCTION RUNS, READ IN THE TOP OF THE SURFACE LAYER
C        AND THE SIGMA OF THE WIND AZIMUTH ANGLE TO BE USED FOR ALL RUNS
C
      CALL DREAD(NAMEF,11,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,33,3,0)
      NIN = 6
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,263.0,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CHAR(0.0,YPOS,0,ILINE,33,0,0)
      CALL CODE(NIN)
      READ (IDATAF,108) TOPLAY
      YPOS = YPOS - 32.0
      CALL DREAD(NAMEF,12,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,43,3,0)
      NIN = 2
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,295.0,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CHAR(0.0,YPOS,0,ILINE,37,0,0)
      CALL CODE(NIN)
      READ (IDATAF,109) ISIGAZ
      IF(NIN .EQ. 1)ISIGAZ = ISIGAZ*10
      SIGAZ = FLOAT(ISIGAZ)
      YPOS = YPOS - 32.0
C
C        NET PLOTS FORMS? ....USE ONLY IF ABSOLUTELY NECESSARY
C
 12   CALL DREAD(NAMEF,24,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,30,3,0)
      CALL CHAR(240.0,YPOS,0,ILINE(16),30,0,0)
      CALL CHAR(472.0,YPOS,0,ILINE(30),4,3,0)
C
C     [illegible] VERSION, CALL  "IN"
```

```
      IF(LVERSN)GO TO 13
C
C  OTHERWISE RESPOND WITH Y FOR YES, N FOR NO TO ANSWER QUESTIONS
C
      READ(LU,90) IANS
      IF(IANS.EQ.1HO)GO TO 81
C
C  "NO" IS THE DEFAULT CASE
C
      IX = 30
C
C  THIS IS THE "YES" RESPONSE
C
      IF(IANS .EQ. YES) IX = 25
      GO TO 14
 13   CONTINUE
      CALL IN(1,JTYPE,0.0,0.0,0,0,0,0,31,0,31,IX,IY)
 14   CONTINUE
      CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)CALL CHAR(472.0,YPOS,0,IERS,5,0,0)
      IF(IX .GT. 25)CALL CHAR(400.0,YPOS,0,IERS,4,0,0)
      MFORM = 1
      IF(IX .GT. 25)MFORM = 2
      YPOS = YPOS - 32.0
C
C        GENERATE NET PLOTS?....
C
      CALL DREAD(NAMEF,23,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,17,3,0)
      CALL CHAR(384.0,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.0,YPOS,0,ILINE(30),6,0,0)
C
C  IF PLASMASCOPE VERSION, SKIP TO THE CALL TO "IN"
C
      IF(LVERSN)GO TO 15
C
C  OTHERWISE RESPOND WITH Y FOR YES, N FOR NO TO ANSWER QUESTIONS
C
      READ(LU,90) IANS
      IF(IANS.EQ.1H0)GO TO 81
C
C  "YES" IS THE DEFAULT CASE
C
      IX=25
C
C  THIS IS THE "NO" RESPONSE
C
      IF(IANS .NE. NO)GO TO 16
      IX = 30
      GO TO 16
 15   CONTINUE
      CALL IN(1,JTYPE,0.0,0.0,0.0,0,0,31,0,31,IX,IY)
 16   CONTINUE
      CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)CALL CHAR(464.0,YPOS,0,IEPS,6,0,0)
      IF(IX .GT. 25)CALL CHAR(384.0,YPOS,0,IEPS,8,0,0)
      MONLY = 0.
      IF(IX .GT. 25)MONLY = 1
      YPOS = YPOS - 72.0
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C        GENERATE PRINTER OUTPUT?
C
      CALL DREAD(NAMEF,25,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,26,3,0)
      CALL CHAR(384.0,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.0,YPOS,0,ILINE(30),6,0,0)
C
C  IF PLASMASCOPE VERSION, SKIP TO THE CALL TO "IN"
C
      IF(LVERSN)GO TO 17
C
C  OTHERWISE RESPOND WITH Y FOR YES, N FOR NO TO ANSWER QUESTIONS
C
      READ(LU,90) IANS
      IF(IANS.EQ.1HO)GO TO 81
C
C "YES" IS THE DEFAULT CASE
      IX=25
C
C  THIS IS THE "NO" RESPONSE
C
      IF(IANS .NE. NO)GO TO 18
      IX=30
      GO TO 18
 17   CONTINUE
      CALL IN(1,JTYPE,0.0,0.0,0,0,0,0,31,0,31,IX,IY)
 18   CONTINUE
      CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)CALL CHAR(464.0,YPOS,0,IERS,6,0,0)
      IF(IX .GT. 25)CALL CHAR(384.0,YPOS,0,IERS,8,0,0)
      IF(IX .GT. 25)LUPRNT = 64
      YPOS = YPOS - 32.0
CBJ
CBJ      READ IN AND WRITE OUT THE LAUNCH TIME AND DATE -- DEFAULT
CBJ      IS CURRENT TIME AND DATE
C
      CALL DREAD(NAMEF,8,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,28,3,0)
      CALL CHAR(384.0,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.0,YPOS,0,ILINE(30),6,0,0)
      CALL GETTD(LTIM,LDAY,LMONTH,LYEAR)
      CALL CODE
      WRITE (IDATAF,106) LTIM,SDT,LDAY,LMONTH,LYEAR
      CALL CHAR(224.0,YPOS,0,IDATAF,20,0,0)
C
C  IF PLASMASCOPE VERSION, SKIP TO THE CALL TO "IN"
C
      IF(LVERSN)GO TO 19
C
C  OTHERWISE RESPOND WITH Y FOR YES, N FOR NO TO ANSWER QUESTIONS
C
      READ(LU,90) IANS
      IF(IANS.EQ.1HO)GO TO 81
C
C  "YES" IS THE DEFAULT CASE
C
      IX=25
```

```
C
C  THIS IS THE "NO" RESPONSE
C
       IF(IANS.NE.NO) GO TO 22
       IX = 30
       GO TO 22
 19    CONTINUE
       CALL IN(1,JTYPE,0.0,0.0,0,0,0,0,31,0,31,IX,IY)
 22    CONTINUE
       CALL CHAR(0.0,YPOS,0,ILINE,28,0,0)
       CALL CHAR(384.0,YPOS,0,ILINE(25),15,0,0)
       IF(IX .LE. 25)CALL CHAR(464.0,YPOS,0,IERS,6,0,0)
       IF(IX .GT. 25)CALL CHAR(384.0,YPOS,0,IERS,8,0,0)
       YPOS = YPOS - 32.0
       IF(IX .LE. 25)GO TO 20
       CALL DREAD(NAMEF,9,ILINE)
       CALL LERS(YPOS,LVERSN)
       CALL CHAR(0.0,YPOS,0,ILINE,26,3,0)
       NIN = 20
       CALL BLANK(IDATAF,10)
       CALL IN(0,JTYPE,207.0,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
       CALL CHAR(0.0,YPOS,0,ILINE,26,0,0)
       CALL CODE(NIN)
       READ (IDATAF,102) (LAUNTD(I),I=1,10)
       YPOS = YPOS - 32.0
       IF(LAUNTD(1) .EQ. BLANKS)GO TO 20
       LTIME = .TRUE.
       CALL CODE(NIN)
       READ (LAUNID,103) LTIM,LDAY,LMONTH,LYEAR
       GO TO 21
    20 LAUNTD(4) = SDT
C
C         READ IN THE LAUNCH VEHICLE, LET IT DEFAULT IF NOT ENTERED,
C         WRITE IT BACK OUT, AND FILL THE VPAR ARRAY WITH THE
C         APPROPRIATE VEHICLE PARAMETERS
C
    21 CALL DREAD(NAMEF,10,ILINE)
       CALL LERS(YPOS,LVERSN)
       CALL CHAR(0.0,YPOS,0,ILINE,24,3,0)
       CALL CHAR(192.0,YPOS,0,ILINE(13),24,0,0)
       CALL CHAR(416.0,YPOS,0,ILINE(27),11,3,0)
C
C*** IF PLASMASCOPE, SKIP TO THE CALL TO "IN"
       IF(LVERSN)GO TO 217
C
C*** OTHERWISE RESPOND WITH D FOR DELTA, T FOR TITAN, OR S FOR SHUTTLE
       READ(LU,90) IANS
       IF(IANS.EQ.1HO)GO TO 81
C
C*** LET A 0 ANSWER (BLANK) DEFAULT TO THE TITAN
CC     IF( (IANS.NE.1HD) .AND. (IANS.NE.1HS) ) IX = 20
       IF( (IANS.NE.1HD) .AND. (IANS.NE.1HT) ) IX = 30
       IF(IANS .EQ. 1HD) IX = 10
CC     IF(IANS .EQ. 1HS) IX = 30
       IF(IANS .EQ. 1HT) IX = 20
       GO TO 218
   217 CONTINUE
       CALL IN(1,JTYPE,0.0,0.0,0,0,0,0,31,0,31,IX,IY)
   218 CONTINUE
       CALL CHAR(0.0,YPOS,0,ILINE,24,0,0)
```

```
      IF(IX .LE. 15)CALL CHAR(312.0,YPOS,0,IERS,24,0,0)
      IF(IX.GT.15 .AND. IX.LT.24)CALL CHAR(192.0,YPOS,0,IERS,12,0,0)
      IF(IX.GT.15 .AND. IX.LT.24)CALL CHAR(416.0,YPOS,0,IERS,11,0,0)
      IF(IX .GE. 24)CALL CHAR(192.0,YPOS,0,IERS,24,0,0)
      YPOS = YPOS - 32.0
      IF(IX .LE. 15)GO TO 25
      IF(IX .LE. 23)GO TO 24
C
C**********************************************************************
C*****       IF IVHICL = 0   THEN IT IS A SHUTTLE LAUNCH.          *******
C**********************************************************************
C
      IVHICL = 0
      GO TO 26
C
C**********************************************************************
C*****       IF IVHICL = 1   THEN IT IS A TITAN LAUNCH.            *******
C**********************************************************************
C
   24 IVHICL = 1
      GO TO 26
C
C**********************************************************************
C*****       IF IVHICL = 2   THEN IT IS A DELTA LAUNCH.            *******
C**********************************************************************
C
   25 IVHICL = 2
   26 I = IVHICL + 1
C
C         FILL THE VPAR ARRAY
C
      DO 28 J=1,18
   28 VPAR(J) = VPARS(J,I)
C
C         CHANGE IN BOTTOM LAYER WITH TOTAL REFLECTION
C                       (COMMENTED-OUT)
C****THE NEXT FEW PAGES OF CODE WERE ORIGINALLY COMMENTED OUT AT THE
C****DIRECTION OF DR. STEPHENS..BUT ARE TEMPORARILLY RESTORED..CAB810825
C*
      CALL DREAD(NAMEF,15,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 16.0
      CALL DREAD(NAMEF,16,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(32.0,YPOS,0,ILINE(3),11,3,0)
      CALL CHAR(160.0,YPOS,0,ILINE(11),43,0,0)
      IF(LVERSN) GO TO 291
      READ(LU,91) IANS
      IF(IANS.EQ.1HO)GO TO 81
      IX = 5
      IF(IANS .EQ. 2HST) IX = 12
      IF(IANS .EQ. 2HSE) IX = 25
      GO TO 293
  291 CONTINUE
      CALL IN(1,JTYPE,0.0,0.0,0,0,0,0,31,0,31,IX,IY)
  293 CONTINUE
      CALL CHAR(0.0,YPOS,0,IERS,64,0,0)
      YP = YPOS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
CBJ
CBJ      CHECK FOR SURFACE -- STABILIZATION -- SOMETHING ELSE
CBJ
         IF(IX .GE. 8)GO TO 29
         ICALC = 0
         CALL CHAR(0.0,YP,0,ILINE,16,0,0)
         GO TO 38
      29 IF(IX .GT. 20)GO TO 30
         ICALC = 1
         CALL CHAR(160.0,YP,0,ILINE(11),16,0,0)
         LAYBOT = 2
         BOTLAY = ALT(LAYBOT)
         CALHT = BOTLAY
         GO TO 38
CBJ
CBJ      DEFAULT HEIGHT OF CALCULATION, Zz?
CBJ
      30 ICALC = 2
         CALL CHAR(320.0,YP,0,ILINE(20),18,0,0)
         CALL DREAD(NAMEF,17,ILINE)
         CALL LERS(YPOS,LVERSH)
         CALL CHAR(0.0,YPOS,0,ILINE,42,3,0)
         CALL CHAR(384.0,YPOS,0,ILINE(25),8,3,0)
         CALL CHAR(464.0,YPOS,0,ILINE(30),6,0,0)
         CALL IN(1,JTYPE,0.0,0.0,0,0,0,0,31,0,31,IX,IY)
         CALL CHAR(0.0,YPOS,0,IERS,42,0,0)
         CALL CHAR(0.0,YPOS,0,ILINE,42,0,0)
         IF(IX .LE. 25)GO TO 37
         CALL CHAR(384.0,YPOS,0,IERS,8,0,0)
         YPOS = YPOS - 32.0
CBJ
CBJ      ENTER HEIGHT OF CALCULATION, Zz
CBJ
         CALL DREAD(NAMEF,18,ILINE)
         CALL LERS(YPOS,LVERSH)
         CALL CHAR(47.0,YPOS,0,ILINE(4),10,3,0)
         NIN = 6
         CALL BLANK(IDATAF,10)
         CALL IN(0,JTYPE,128.0,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
         CALL CODE(NIN)
         READ (IDATAF,*) CALHT
         CALL CHAR(0.0,YPOS,0,IERS,16,0,0)
         CALL CHAR(47.0,YPOS,0,ILINE(4),10,0,0)
         YPOS = YPOS - 32.0
CBJ
CBJ      ENTER SURFACE REFLECTION?
CBJ
         CALL DREAD(NAMEF,19,ILINE)
         CALL LERS(YPOS,LVERSH)
         CALL CHAR(0.0,YPOS,0,ILINE,45,3,0)
         CALL CHAR(383.0,YPOS,0,ILINE(25),8,3,0)
         CALL CHAR(472.0,YPOS,0,ILINE(30),6,0,0)
         CALL IN(1,JTYPE,0.0,0.0,0,0,0,0,31,0,31,IX,IY)
         CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
         IF(IX .LE. 25)REFLEC = 1.0
         IF(IX .LE. 25)CALL CHAR(464.0,YPOS,0,IERS,6,0,0)
         IF(IX .LE. 25)GO TO 34
         CALL CHAR(384.0,YPOS,0,IERS,8,0,0)
         YPOS = YPOS - 32.0
CBJ
```

```
CBJ***THIS SECTION HAS BEEN PLACE BELOW AT 38
CBJ
CBJ      WRITE OUT RF VALUES FOR SELECTION
CBJ
C*      CALL DREAD(NAMEF,20,ILINE)
C*      CALL LERS(YPOS,LVERSN)
C*      CALL CHAR(0.0,YPOS,0,ILINE,64,3,0)
C*      NIN = 4
C*      CALL BLANK(IDATAF,10)
C*      CALL IN(2,JTYPE,440.0,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
C*      IF(JTYPE .NE. 0)GO TO 31
C*      CALL CODE(NIN)
C*      READ (IDATAF,*) REFLEC
C*      GO TO 32
C* 31   IX = IX/2
C*      IF(IX .EQ. 1)REFLEC = 0.8
C*      IF(IX .EQ. 3)REFLEC = 0.7
C*      IF(IX .EQ. 5)REFLEC = 0.5
C*      IF(IX .EQ. 7)REFLEC = 0.3
C*      IF(IX .EQ. 9)REFLEC = 0.1
C*      IF(IX .EQ. 11)REFLEC = 0.0
C*      CALL CODE
C*      WRITE (IDATAF,110) REFLEC
C* 32   CALL CHAR(0.0,YPOS,0,IERS,64,0,0)
C*      CALL CHAR(48.0,YPOS,0,ILINE,6,0,0)
C*      CALL CHAR(88.0,YPOS,0,IDATAF,4,0,0)
C*      IF(JTYPE .NE. 0)GO TO 34
C*      CALL CODE(4)
C*      READ (IDATAF,*) REFLEC
C* 34   YPOS = YPOS - 32.0
CBJ
CBJ      DEFAULT HEIGHT OF BASE LAYER?
CBJ
        CALL DREAD(NAMEF,21,ILINE)
        CALL LERS(YPOS,LVERSN)
        CALL CHAR(0.0,YPOS,0,ILINE,46,3,0)
        CALL CHAR(384.0,YPOS,0,ILINE(25),8,3,0)
        CALL CHAR(464.0,YPOS,0,ILINE(30),6,0,0)
        CALL IN(1,JTYPE,0.0,0.0,0,0,0,0,31,0,31,IX,IY)
        CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
        IF(IX .LE. 25)CALL CHAR(464.0,YPOS,0,IERS,6,0,0)
        IF(IX .LE. 25)GO TO 36
        CALL CHAR(384.0,YPOS,0,IERS,8,0,0)
        YPOS = YPOS - 32.0
        CALL DREAD(NAMEF,22,ILINE)
        CALL LERS(YPOS,LVERSN)
        CALL CHAR(47.0,YPOS,0,ILINE(4),10,3,0)
CBJ
CBJ      ENTER HEIGHT OF BASE LAYER
CBJ
        NIN = 6
        CALL BLANK(IDATAF,10)
        CALL IN(0,JTYPE,144.0,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
        CALL CODE(NIN)
        READ (IDATAF,*) BOTLAY
        CALL CHAR(47.0,YPOS,0,ILINE(4),10,0,0)
        YPOS = YPOS - 32.0
        GO TO 38
     36 BOTLAY = 0.0
        GO
```

```
   37 CALL CHAR(0.0,YPOS,0,IERS,64,0,0)
      CALL CHAR(0.0,YPOS,0,ILINE,53,0,0)
      YPOS = YPOS - 32.0
CBJ*** THIS IS THE END OF THE COMMENTED-OUT SECTION
CBJ
   38 CONTINUE
CBJ
CBJ*** INSERTED SECTION ON SURFACE LAYER REFLECTION
CBJ
      IF(IRUN.NE.1  GO TO 34
      YPOS = YPOS - 32.0
CBJ
CBJ      ENTER SURFACE REFLECTION?
CBJ
      CALL DREAD(NAMEF,19,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,45,3,0)
      CALL CHAR(383.0,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(472.0,YPOS,0,ILINE(30),6,0,0)
      CALL IN(1,JTYPE,0.0,0.0,0,0,0,0,31,0,31,IX,IY)
      CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)REFLEC = 1.0
      IF(IX .LE. 25)CALL CHAR(464.0,YPOS,0,IERS,6,0,0)
      IF(IX .LE. 25)GO TO 34
      CALL CHAR(384.0,YPOS,0,IERS,8,0,9)
      YPOS = YPOS - 32.0
CBJ
CBJ      WRITE OUT RF VALUES FOR SELECTION
CBJ
      CALL DREAD(NAMEF,20,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,64,3,0)
      NIN = 4
      CALL BLANK(IDATAF,10)
      CALL IN(2,JTYPE,440.0,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      IF(JTYPE .NE. 0)GO TO 31
      CALL CODE(NIN)
      READ (IDATAF,*) REFLEC
      GO TO 32
   31 IX = IX/2
      IF(IX .EQ. 1)REFLEC = 0.8
      IF(IX .EQ. 3)REFLEC = 0.7
      IF(IX .EQ. 5)REFLEC = 0.5
      IF(IX .EQ. 7)REFLEC = 0.3
      IF(IX .EQ. 9)REFLEC = 0.1
      IF(IX .EQ. 11)REFLEC = 0.0
      CALL CODE
      WRITE (IDATAF,110) REFLEC
   32 CALL CHAR(0.0,YPOS,0,IERS,64,0,0)
      CALL CHAR(48.0,YPOS,0,ILINE,6,0,0)
      CALL CHAR(88.0,YPOS,0,IDATAF,4,0,0)
      IF(JTYPE .NE. 0)GO TO 34
   34 YPOS = YPOS - 32.0
C
C        DO LOOP ON THE RUN NUMBER
C
      DO 79 RUNNUM=1,NUMRUN
C
C        SET UP THE FILE NAME FOR THIS RUN. GET ...
C        ... WRITE OUT THE HEADER
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
      I = RUNNUM + IFOFF
      CALL CODE
      WRITE (J,109) I
      CALL B2Z(J,FILE(3))
      CALL GETTD(ITIME,IDAY,IMONTH,IYEAR)
      I = IVHICL + 1
      WRITE (LUPRNT,200) RUNNUM,(FILE(J),J=1,3),(VNAMES(J,I),J=1,4)
      IF(LTIME)WRITE (LUPRNT,201) LTIM,LAUNTD(3),LAUNTD(4),LDAY,
     .                              LMONTH(1),LMONTH(2),LYEAR
      WRITE (LUPRNT,202) ITIME,LAUNTD(4),IDAY,IMONTH,IYEAR
C
C        IF THE DATA IS ON A DISK FILE, READ FROM DISK -- IF IT
C        IS ON TAPE, READ IT AS KSC 1965 DATA IN SUBROUTINE KSC65
C
      IF(IPLACE .NE. 2 .AND. IPLACE .NE. 4) GO TO 39
      CALL KSC65(IBUF,IALT,DPTEMP,IFOFF1,IEOF,ITIMG,IDAYG,IMONG,ISNDG,
     .          RUNNUM)
      IFOFF1 = 1
      IF(IEOF .EQ. 1)GO TO 81
      IF(IEOF .EQ. 2)GO TO 79
      GO TO 48
C
C        OPEN THE DATA FILE FOR THIS RUN
C
   39 CALL OPEN(IDCB,IERR,FILE,0,0,0,272)
      IF(IERR .GE. 0)GO TO 40
      WRITE (LUPRNT,203) IERR
      GO TO 79
C
C        READ THE HEADINGS FROM THE DATA FILE, SETTING UP THE
C        APPROPRIATE PARAMETERS
C
   40 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .GE. 0)GO TO 42
   41 WRITE (LUPRNT,204) IERR
      GO TO 79
   42 IF(IBUF(1) .NE. TE)GO TO 40
   43 WRITE (LUPRNT,205) (IBUF(I),I=1,LEN)
      CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .LT. 0)GO TO 41
      IF(IBUF(1).NE.RA .AND. IBUF(1).NE.FO)GO TO 43
      ISNDFO = 0
      IF(IBUF(1) .EQ. FO)ISNDFO = 1
      WRITE (LUPRNT,205) (IBUF(I),I=1,LEN)
      CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .LT. 0)GO TO 41
      WRITE (LUPRNT,205) (IBUF(I),I=1,LEN)
C
C        READ THE SOUNDING/FORECAST TIME
C
      CALL READF(IDCB,IERR,IBUF,9,LEN)
      IF(IERR .LT. 0)GO TO 41
      CALL CODE(18)
      READ (IBUF,104) ISTIME,ISDAY,ISMON(1),ISMON(2),ISYEAR
C
C        CHANGE TO EST OR EDT DEPENDING ON LAUNCH TIME
C
      ISTIME = ISTIME - 500
      IF(IPLACE .EQ. 1)ISTIME = ISTIME + 100
```

```
      IF(LAUNID(4) .NE. 2HST)ISTIME = ISTIME + 100
      IF(ISTIME .GT. 0)GO TO 44
      ISTIME = 2400 + ISTIME
      ISDAY = ISDAY - 1
C
C        WRITE OUT THE NEXT LINE OF THE HEADER
C
   44 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .LT. 0)GO TO 41
      WRITE (LUPRNT,205) (IBUF(I),I=1,LEN)
C
C        WRITE OUT THE SOUNDING/FORECAST TIME
C
      WRITE (LUPRNT,206) ISTIME,ITIMEZ,LAUNID(4),ISDAY,ISMON(1),
     *                   ISMON(2),ISYEAR
C
C        FIND THE FIRST DATA POINT WITH AN ALTITUDE OF 10 FEET
C        OR ABOVE
C
   45 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR .LT. 0)GO TO 41
      CALL B2Z(IBUF,J)
      IF(J.LT.ZERO0 .OR. J.GT.ZERO1)GO TO 45
      CALL CODE(2*LEN)
      READ (IBUF,105) IALT(1),IDIR(1),SPEED(1),TEMP(1),DPTEMP(1),
     *               PRESS(1),SUPDEN
      IF(IALT(1) .LT. 10)GO TO 45
C
C        TRY TO FIND A TOTAL OF 30 DATA POINTS WITH ALTITUDES
C        BETWEEN 20 FT AND 10,000 FT INCLUSIVE
C
      NUM = 1
      DO 47 I=2,30
   46 CALL READF(IDCB,IERR,IBUF,40,LEN)
      IF(IERR.LT.0 .AND. IERR.NE.-12)GO TO 41
      IF(LEN .EQ. -1)GO TO 48
      CALL B2Z(IBUF,J)
      IF(J.LT.ZERO0 .OR. J.GT.ZERO1)GO TO 46
      CALL CODE(2*LEN)
      READ (IBUF,105) IALT(I),IDIR(I),SPEED(I),TEMP(I),DPTEMP(I),
     *               PRESS(I)
      IF(IALT(I).LT.20 .OR. IALT(I).GT.10000)GO TO 46
   47 NUM = I
C
C        ZERO OUT THE REMAINING ELEMENTS OF THE ARRAYS
C
   48 NUM1 = NUM + 1
      IF(NUM1 .GT. 30)GO TO 51
      DO 49 I=NUM1,30
      ALT(I) = 0.0
      IDIR(I) = 0
      SPEED(I) = 0.0
      TEMP(I) = 0.0
      DPTEMP(I) = 0.0
   49 PRESS(I) = 0.0
C
C        CONVERT TO METRIC UNITS
C
   51 DO 52 I=1,NUM
      ALT(I) = 0.3048 * FLOAT(IALT(I))
```

```
52 SPEED(I) = 0.515 * SPEED(I)
C
C       SORT ALL THE DATA POINTS SO THEY APPEAR IN ASCENDING
C       ORDER OF ALTITUDE
C
   NUM1 = NUM - 1
   DO 58 I=1,NUM1
   JJ = NUM - I
   DO 57 J=1,JJ
   J1 = J + 1
   IF(ALT(J) .LE. ALT(J1))GO TO 57
   ARG = ALT(J)
   ALT(J) = ALT(J1)
   ALT(J1) = ARG
   IARG = IDIR(J)
   IDIR(J) = IDIR(J1)
   IDIR(J1) = IARG
   ARG = SPEED(J)
   SPEED(J) = SPEED(J1)
   SPEED(J1) = ARG
   ARG = TEMP(J)
   TEMP(J) = TEMP(J1)
   TEMP(J1) = ARG
   ARG = DPTEMP(J)
   DPTEMP(J) = DPTEMP(J1)
   DPTEMP(J1) = ARG
   ARG = PRESS(J)
   PRESS(J) = PRESS(J1)
   PRESS(J1) = ARG
57 CONTINUE
58 CONTINUE
C
C       CALCULATE THE POTENTIAL TEMPERATURE
C
   DO 62 I=1,NUM
62 PTEMP(I) = (TEMP(I) + 273.16) * ((1000.0/PRESS(I))**0.288)
C
C       WRITE THE HEADER FOR SOUNDING OR FORECAST
C
   IF(ISNDFO .EQ. 1)GO TO 64
   WRITE (LUPRNT,207)
   GO TO 65
64 WRITE (LUPRNT,208)
C
C       WRITE THE SURFACE DENSITY AND ALL THE DATA POINTS
C
65 WRITE (LUPRNT,209) SURDEN
   WRITE (LUPRNT,210)
   DO 68 I=1,NUM
   IALTF = 3.281 * ALT(I) + 0.5
   IALTM = ALT(I) + 0.5
   APTEMP = PTEMP(I) - 273.16
68 WRITE (LUPRNT,211) I,IALTF,IALTM,IDIR(I),SPEED(I),TEMP(I),
  .                APTEMP,DPTEMP(I),PRESS(I)
C
C       CLOSE THE DATA FILE
C
   CALL CLOSE(IDCB)
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
      CALL NGRAF
      CALL RWDIS(NAME,1)
      CALL EXEC(9,RCLDR)
      CALL RWDIS(NAME,0)
      CALL GRAF(1)
C
C        PROCESS THE NEXT RUN
C
   79 CONTINUE
C
C        TERMINATE OF PROCESS MORE DATA?
C
   81 CALL DREAD(NAMEF,14,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.,YPOS,0,ILINE,24,3,0)
      CALL CHAR(224.,YPOS,0,ILINE(15),14,3,0)
      CALL CHAR(352.,YPOS,0,ILINE(23),12,0,0)
C
C     TEST FOR RUNNING ON PLASMASCOPE OR CRT; LVERSN IS TRUE
C     IF RUNNING ON THE PLASMASCOPE...THEN TEST FOR RETURN (P),
C     WHICH WILL PROCESS MORE DATA, OR TERMINATE (T) WHICH
C     WILL STOP EXECUTION OF REED.
C
      IF(LVERSN) GO TO 881
      READ(LU,90) IANS
      IX = 15
      IF(IANS.NE.1HR) GO TO 882
      IX = 25
      GO TO 882
  881 CONTINUE
      CALL INC      ,0.,0.,0.0,0,0.31,0.31,IX,IY)
  882 CONTINUE
      .F(IX .LI      GO TO 82
C
C        PROCESS MORE DA
C
      CALL LERS(YPOS,LVERS
      CALL CHAR(0.,YPOS,0      ,28,0,0)
      CALL CHAR(352.,YPOS,0,ILINE(23),12,0,0)
CC    YPOS = 458.
      GO TO 2
C
C        TERMINATE EXECUTION -- WRITE OUT A MESSAGE
C
   82 CALL DREAD(NAMEF,13,ILINE)
      CALL LERS(YPOS,LVERSN)
      YPOS = YPOS - 32.
      CALL CHAR(0.,YPOS,0,ILINE,64,3,0)
C
C        DELAY 5 SECONDS BEFORE CLEARING SCOPE
C
      CALL WAIT(3,2)
CBJ
CBJ ***  PURGE THE DISK FILE USED FOR STORING COMMON BLOCK DATA
C
      CALL PURGE(IDCB,IERR,NAME,0,0)
C
C        REINITIALIZE SCOPE FOR FULL OPERATION
C
```

```
CC     CALL CLEAR
       IF(LVERSN) CALL CLEAR
       CALL NGRAF
C
C         OF ALL SEGMENTS -- IF ERRORS ENCOUNTERED, WRITE OUT A MESSAGE
C         AND TERMINATE REEDS
C
       DO 94 I=1,NUMSEG
       CALL IDRPD(NAMRP(1,I),IERR)
       IF(IERR .GE. 0)GO TO 94
       WRITE (LU,213) IERR
       STOP
    94 CONTINUE
C
C         TERMINATE REEDS
C
       STOP
C
       END
       END$
```

&PREEDT T-00003 IS ON CR00003 USING 00141 BLKS R=6000

```
FTN4X,L
      SUBROUTINE RUDIS(NAME,JJ),                          *REEDS,CA8810315
C
C     ------------------------------------------------------------
C     -                                                          -
C     -   THIS SUBROUTINE READS AND WRITES COMMON AREAS TO DISC  -
C     -   TO BE UTILIZED BY THE VARIOUS REED PROGRAMS.           -
C     -                                                          -
C     ------------------------------------------------------------
C
C >>>   COMMON AREAS
C
C
C >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISHDFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,ICOTMP,
     .       NUMFUN,LU,NUM,MONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMONK(2),ISYEAR
C
C >>>   LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME
C
C >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPIZ
C
C >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCFX,DOSMAX,DOSFX,RNGSMC,RNGSMD
C
C >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QL3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATH,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PHO,VPAR(17)),(GAMMAX,VPAR(18))
      INTEGER DUCB(144),OG?F(557
```

9 15
17 27
19 20
184 335
5 5
5 5
14 14
43 9

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DIMENSION NAME(3) , ISIZE(2)
      EQUIVALENCE (OBUF(1),PI)

CBJ *** ISIZE(1) = # BLOCKS EXPECTED TO BE WRITTEN READ DURING RUN
CBJ *** ISIZE(2) = # WORDS IN THE COMMON BLOCK TO BE TRANSFERRED
      DATA ISIZE/30,557/

CBJ *** IF THE SCRATCH DATA FILE (i.e. *REED2) ISN'T THERE, CREATE IT
CBJ *** B E W A R E !!  BE SURE FILE IS PURGED BEFORE STARTING RUN !!
      CALL OPEN(ODCB,IERR,NAME,0)
      IF( (IERR.LT.0) .AND. (IERR.NE.-6) ) WRITE(6,770) IERR
770   FORMAT(" *** ATTEMPT TO OPEN, IERR =",I4)
      IF(IERR.LT.0)
     .   CALL CREAT(ODCB,IERR,NAME,ISIZE,3,0,0,144)
      IF(IERR.LE.0) WRITE(6,799) IERR
799   FORMAT(" *** CREATE/OPEN, IERR=",I4)

CBJ *** READ OR WRITE, AS PER ORDER
      IF(JJ.EQ.1)CALL WRITF(ODCB,IERR,OBUF,ISIZE(2))
      IF(JJ.EQ.0)CALL READF(ODCB,IERR,OBUF,ISIZE(2))
      CALL CLOSE(ODCB,IERR)
      RETURN
      END
      SUBROUTINE KSC65(IBUF,IALT,DPTEMP,IWANT,IEOF,ITIMG,IDAYG,IMONG,
     .                 ISHDG,RUNNUM)
C
C   -----------------------------------------------------------------
C   -                                                               -
C   -     THIS SUBROUTINE READS IN DATA FOR THE REED DIFFUSION      -
C   -     MODEL FROM MAG TAPE IN KSC 1965 FORMAT                    -
C   -                                                               -
C   -----------------------------------------------------------------
C
C
C  >>>    COMMON AREAS
C
C
C  >>>    MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>    VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C  >>>    OPTION PARAMETERS
      COMMON YPOS,YES,ISHOFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,MONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>    MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C  >>>    CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C  >>>    SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C  >>>    LAUNCH TIME
```

```
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME
C
C >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOFLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SOSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
C** DIMENSION STATMENTS
C
      DIMENSION IMONG(5,2),ITIMG(5),IDAYG(5)
C
C** TYPE STATMENTS
C
      INTEGER RUNNUM
C
C        INPUT FORMAT STATEMENTS
C
 1000 FORMAT (40A2)
 1001 FORMAT (I4,3XI2,1XA2,A1,1XI4)
 1002 FORMAT (1XI6,3XI3,5XF3.0,2XF5.1,3XF5.1,3XF6.1,15XF6.1)
C
C        OUTPUT FORMAT STATEMENT
C
 2000 FORMAT ("0"5X"TIME: "I4,1XA1,A2,4X"DATE: "I2,1XA2,A1,1XI4)
C
C        DIMENSION STATEMENT
C
      DIMENSION IBUF(1),IALT(1),DPTEMP(1)
C
C        INITIALIZE THE COUNTER FOR THE NUMBER OF SETS OF DATA TO 0
C
      IGOT = 0
C
C        READ DATA FROM TAPE
C
    4 READ (8,1000) (IBUF(I),I=1,40)
C
C        IF AN EOF ON TAPE, SET THE EOF FLAG AND RETURN
C
      CALL EXEC(13,8,IEOTS)
      IEOF = IAND(ISHIF(IEOTS,-7),1)
      IF(IEOF .EQ. 1)RETURN
C
C        KEEP READING UNTIL THE STANDARD LEVEL DATA IS FOUND
C
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      IF(IBUF(2) .NE. 2HST)GO TO 4
    7 READ (8,1000) (IBUF(I),I=1,40)
      IF(IBUF(1).NE.2HCA .OR. IBUF(2).EQ.2HST)GO TO 7
C
C        READ THE SOUNDING/FORECAST TIME
C
      READ (8,1001) ISTIME,ISDAY,ISMON(1),ISMON(2),ISYEAR
C
C** CHECK FOR SPECIFIED SOUNDING TIME AND DATE
C
      IF(ISNDG.EQ.0) GO TO 8
      IF(ISTIME.EQ.ITIMG(RUNNUM).AND.ISDAY.EQ.IDAYG(RUNNUM).AND.
     .   ISMON(1).EQ.IMONG(RUNNUM,1).AND.ISMON(2).EQ.IMONG(RUNNUM,2))
     .   GO TO 8
      GO TO 4
C
C        CHANGE TO EST OR EDT DEPENDING ON LAUNCH TIME
C
8     CONTINUE
      ISTIME = ISTIME - 500
      IF(IPLACE .EQ. 1)ISTIME = ISTIME - 300
      IF(LAUNTD(4) .NE. 2HST)ISTIME = ISTIME + 100
      IF(ISTIME .GT. 0)GO TO 11
      ISTIME = 2400 + ISTIME
      ISDAY = ISDAY - 1
C
C        FIND THE KEY WORD ALTITUDE
C
   11 READ (8,1000) (IBUF(I),I=1,40)
      IF(IBUF(2) .EQ. 2HST)GO TO 7
      IF(IBUF(1) .NE. 2HAL)GO TO 11
C
C        LIMIT DATA TO 30 POINTS -- READ THE STANDARD LEVEL DATA
C
      DO 19 I=1,30
   15 READ (8,1002) IALT(I),IDIR(I),SPEED(I),TEMP(I),DPTEMP(I),PRESS(I),
     .                 SURDN
      IF(SPEED(I).EQ.999.0 .OR. IDIR(I).EQ.999)GO TO 15
      IF(IDIR(I) .EQ. 360)IDIR(I) = 0
      IF(I .EQ. 1)SURDEN = SURDN
      IF(IALT(I) .GT. 10000)GO TO 22
   19 CONTINUE
   22 NUM = I
      IF(NUM .GT. 30)GO TO 34
C
C        FIND THE KEY WORD MANDATORY
C
   25 READ (8,1000) (IBUF(I),I=1,40)
      IF(IBUF(2) .EQ. 2HST)GO TO 7
      IF(IBUF(10).NE.2HOR .AND. IBUF(15).NE.2HOR)GO TO 25
      READ (8,1000) I
C
C        LIMIT DATA TO 30 POINTS -- READ THE MANDATORY LEVEL DATA
C
      DO 29 I=NUM,30
   27 READ (8,1002) IALT(I),IDIR(I),SPEED(I),TEMP(I),DPTEMP(I),PRESS(I)
      IF(SPEED(I).EQ.999.0 .OR. IDIR(I).EQ.999)GO TO 27
      IF(IDIR(I) .EQ. 360)IDIR(I) = 0
      IF(IALT(I) .GT. 10000)GO TO 34
   29 CONTINUE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C        NUM IS THE NUMBER OF DATA POINTS
C
   34 NUM = I - 1
C
C        INCREMENT THE COUNTER -- IF THIS IS THE SET OF DATA DESIRED,
C        WRITE OUT THE SOUNDING/FORECAST TIME -- OTHERWISE GET THE NEXT
C        SET
C
      IGOT = IGOT + 1
      IF(IGOT .LT. IWANT)GO TO 4
C
C        WRITE OUT THE SOUNDING/FORECAST TIME
C
      WRITE (LUPRNT,2006) ISTIME,ITIMEZ,LAUNTD(4),ISDAY,ISMON(1),
     .                    ISMON(2),ISYEAR
C
C        THERE MUST BE 5 OR MORE DATA POINTS FOR THIS TO BE A VALID SET
C        OF DATA -- IF THERE IS NOT, RETURN WITH IEOF=2
C
      IF(NUM .GE. 5)RETURN
      IEOF = 2
      RETURN
C
C        END OF KSC65
C
      END
      SUBROUTINE GETTD(ITIME,IDAY,IMONTH,IYEAR)
C
C  -----------------------------------------------------------------
C  -                                                               -
C  -  THIS SUBROUTINE RETURNS THE CURRENT TIME, DAY, IMONTH, AND YEAR  -
C  -                                                               -
C  -----------------------------------------------------------------
C
C
C        TYPE AND DIMENSION STATEMENTS
C
      INTEGER DAYMON(12)
      DIMENSION IMONTH(2),MONTHS(2,12),IT(5)
C
C        DATA STATEMENTS
C
      DATA MONTHS/2HJA,1HN,2HFE,1HB,2HMA,1HR,2HAP,1HR,
     .            2HMA,1HY,2HJU,1HN,2HJU,1HL,2HAU,1HG,
     .            2HSE,1HP,2HOC,1HT,2HNO,1HV,2HDE,1HC/
      DATA DAYMON/31,28,31,30,31,30,31,31,30,31,30,31/
C
C        CALL EXEC TO RETURN CURRENT TIME, JULIAN DAY, AND YEAR
C
      CALL EXEC(11,IT,IYEAR)
C
C        USE JUST HOURS AND MINUTES FOR THE TIME
C
      ITIME = 100 * IT(4) + IT(3)
C
C        MAKE APPPOPRIATE ADJUSTMENTS IF THIS IS A LEAP YEAR
C
      DAYMON(2) = 28
      ? = IYEAR/4
```

```
      IF(4*I .EQ. IYEAR)DAYMON(2) = 29
C
C        CONVERT THE JULIAN DAY INTO A MONTH AND A DAY
C
      IDAY = IT(5)
      DO 7 I=1,12
      IDAY = IDAY - DAYMON(I)
      IF(IDAY .LE. 0)GO TO 12
    7 CONTINUE
   12 IDAY = IDAY + DAYMON(I)
      IMONTH(1) = MONTHS(1,I)
      IMONTH(2) = MONTHS(2,I)
C
C        RETURN TO THE CALLING PROGRAM
C
      RETURN
C
C        END OF GETTD
C
      END
      SUBROUTINE B2Z(IA,IB)
C
C   -------------------------------------------------------------
C   -                                                           -
C   -  THIS SUBROUTINE CHANGES BLANK FILLED WORDS TO ZEROS.     -
C   -                                                           -
C   -------------------------------------------------------------
C
      IB = IAND(IA,177400B)
      IF(IB .EQ. 020000B)IB = 030000B
      IC = IAND(IA,000377B)
      IF(IC .EQ. 000040B)IC = 000060B
      IB = IOR(IB,IC)
      RETURN
      END
      SUBROUTINE DREAD(NAMEF,LNUM,ILINE)
C
C   ------------------------------------------------------------------------
C   -                                                                      -
C   -  THIS SUBROUTINE READS A SPECIFIED QUESTION FROM A SPECIFIED         -
C   -  QUESTION FILE. THE QUESTION IS THEN DISPLAYED UPON THE              -
C   -  FLASHASCOPE FOR PROCESSING.                                         -
C   -                                                                      -
C   ------------------------------------------------------------------------
C
      DIMENSION NAMEF(3),IDCB(144),IBUF(40),ILINE(32)
      CALL OPEN(IDCB,IERR,NAMEF,0)
      LOOP = LNUM - 1
      DO 10 I=1,LOOP
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
   10 CONTINUE
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,100) (ILINE(I),I=1,32)
  100 FORMAT(32A2)
      CALL CLOSE(IDCB,IERR)
      RETURN
      END
```

```
      SUBROUTINE BLANK(IBUF,II)
C
C ----------------------------------------------------------------
C -                                                              -
C -  THIS SUBROUTINE FILLS A SPECIFIED ARRAY WITH A SPECIFIED    -
C -  NUMBER OF BLANKS.                                           -
C -                                                              -
C ----------------------------------------------------------------
C
      DIMENSION IBUF(40)
      DATA IBLK/2H  /
      DO 10 I=1,II
10    IBUF(I) = IBLK
      RETURN
      END
      SUBROUTINE LERS(YPOS,LVERSN)
C
C ----------------------------------------------------------------
C -                                                              -
C -  THIS SUBROUTINE ERASES A SPECIFIED LINE OF 64-CHARACTERS    -
C -  FROM THE PLASMASCOPE.                                       -
C -                                                              -
C ----------------------------------------------------------------
C
      DIMENSION IERS(32)
      DATA IERS/32*2H  /
      IF(YPOS .GT. 40.0)GO TO 4
      IF(LVERSN.EQ.0)GO TO 4
      YPOS = 450.0
      CALL CLEAR
      RETURN
    4 CALL CHAR(0.0,YPOS,0,IERS,64,0,0)
      CALL CHAR(0.0,YPOS-16.0,0,IERS,64,0,0)
      RETURN
      END
      SUBROUTINE TOWER(LU,NDAT,IVHICL,NRUN)
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C++                                                     ++
C++ THIS PROGRAM CREATES A DATA FILE ON DISC FOR ADDING ++
C++ TOWER DATA TO THE REED PROGRAMS. THE FORMAT IS      ++
C++ GIVEN IN TM-73337.                                 ++
C++                                                     ++
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C DIMENSION AND DATA STATEMENTS
C
      DIMENSION IPAR(5),NDCB(144),NBUF(59),NAM2(3),LINE1(4),
     .          LINE2(11),LINE3(12),LINE4(58),LINE5(53),LINE6(34),
     .          LEVEL(7),IWD(7),IWS(7),IGS(7),LEVL2(2),ITEM(2),IDF(2),
     .          NDAT(3)
      DATA IBLKS/2H  /
      DATA LINE1/2HTE,2HST,2H N,2HBR/
      DATA LINE2/2HWI,2HND,2H 9,2HYS,2HTE,2HM ,2HTO,2HWE,2HR ,2HDA,2HTA/
      DATA LINE3/2HCA,2HPE,2H C,2HAN,2HAV,2HER,2HAL,2H A,2HFS,2H, ,2HFL,
     .           2HA./
      DATA LINE4/2HYR,2H M,2HON,2H D,2HAY,2H T,2HIM,2HE ,2HIN,2HT ,
     .           2HTW,2HR ,2H  ,2H12,2H F,2HT ,2H  ,2H  ,2H  ,2H54,
     .           2H F,2HT ,2H  ,2H  ,2H  ,2H16,2H2 ,2HFT,2H  ,2H  ,
     .           2H  ,2H20,2H4 ,2HF',2H  ,2H  ,2H6F,2HT ,2HFT,2HT ,
     .           2H54,2HFT,2H  ,2H ^,2HPS,2HF ,2H  ,2H5 ,2H  ,2H 2,
```

4-29

```
     .              2H5 ,2H D,2HIR,2H  ,2H54,2H F,2HT ,2H  /
      DATA LINE5/2H  ,2H  ,2H  ,2H  ,2H  ,2H(Z,2H) ,2H  ,2H  ,
     .              2HNB,2HR ,2HDI,2HR ,2HSP,2HD ,2HGS,2HT ,2HDI,2HP ,
     .              2HSP,2HD ,2HGS,2HT ,2HDI,2HR ,2HSP,2HD ,2HGS,2HT ,
     .              2HDI,2HR ,2HSP,2HD ,2HGS,2HT ,2H T,2HT ,2H D,2HP ,
     .              2H D,2HP ,2H  ,2HRA,2HTE,2H  ,2H P,2HPM,2H  ,2HPP,
     .              2HM ,2H D,2HEV,2H  ,2HTT,2H  ,2HRH,2H  /
      DATA LINE6/2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,
     .              2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H  ,2H(2,
     .              2H95,2H) ,2H  ,2H  ,2H  ,2H(3,2H94,2H) ,2H  ,2H  ,
     .              2H  ,2H(4,2H92,2H) /
      DATA LEVEL/12,54,162,204,295,394,492/
      DATA LEVL2/6,54/
      DATA NAM2/2HTO,2HWE,2HRS/
C
C FORMAT STATEMENTS
C
  100 FORMAT(/" THE NAME OF THE FILE TO BE USED (in:AAAAAA) _")
  101 FORMAT(" IERR =",I4)
  102 FORMAT(/" THE TEST NUMBER (in:XXXXX) _")
  103 FORMAT(11A2)
  104 FORMAT(4A2,1X,I6,1X)
  105 FORMAT(12A2)
  106 FORMAT(58A2)
  108 FORMAT(34A2)
  109 FORMAT(/" IS THE DATA ENTERED CORRECTLY SO FAR? (&dJY=YES&d@"
     .        ",N=NO,S=STOP TOWER) _")
  110 FORMAT(" THE YEAR (in:XX) _")
  111 FORMAT(" THE INTEGER REPRESENTING THE MONTH (in:XX) _")
  112 FORMAT(" THE DAY OF THE MONTH (in:XX) _")
  113 FORMAT(" THE TIME (Z) (in:XXXX) _")
  114 FORMAT(" THE TIME INTERVAL (MIN in:XX) _")
  115 FORMAT(/" THE TOWER NUMBER (in:XXX) _")
  116 FORMAT(5X,"WIND DIRECTION (DEG in:XXX) _")
  117 FORMAT(5X,"WIND SPEED (KTS in:XX) _")
  118 FORMAT(5X,"GUST SPEED (KTS in:XX) _")
  119 FORMAT(/" THE",1X,I2,1X,"FOOT TEMPERATURE (DEG F in:XXX) _")
  120 FORMAT(" THE",1X,I2,1X,"FOOT DEW POINT (DEG F in:XXX) _")
  121 FORMAT(/" THE LAPSE RATE (DEG F in:XX.X) _")
  122 FORMAT(" THE 5 PPM DISTANCE (MI in:XX.X) _")
  123 FORMAT(" THE 25 PPM DISTANCE (MI in:XX.X) _")
  124 FORMAT(" THE STANDARD DEVIATION OF THE WIND AZIMUTH ANGLE"
     .        " (DEG in:XX.X) _")
  125 FORMAT(" THE RELATIVE HUMIDITY AT 54 FEET (% in:XXX) _")
  126 FORMAT(2(I2,2X),I2,1X,I4,2X,I2,2(1X,I3),3(2X,I2,2X,I2,1X,I3),
     .     2(2X,I2),1X,I3,2(1X,I3),3X,F4.1,2X,2(F4.1,1X),F4.1,2(1X,I3))
  127 FORMAT(/" DATA FILE,",1X,3A2,1X,",IS READY")
  128 FORMAT(36X,3(I3,2X,I2,2X,I2,1X),42X)
  129 FORMAT(/" IS THERE DATA FOR ANOTHER VEHICLE USING THE SAME TIME?"
     .        1X,"(Y=YES,&dJN=NO&d@) _")
  130 FORMAT(/1X,I3,1X,"FOOT LEVEL DATA:")
  131 FORMAT(/" THE DATA ENTERED IS: YEAR MONTH DAY TIME INT"/
     .        24X,I2,4X,I2,2X,I2,1X,I4,1X,I3)
  132 FORMAT( 5X,I3,8X,I3,5X,I2,5X,I2)
  133 FORMAT(/" THE TEMPERATURES AND DEW POINTS ENTERED ARE:"/
     .        5X"ALTITUDE TEMPERATURE DEW POINT"/
     .        10X,I3,9X,I3,7X,I3/10X,I3,9X,I3,7X,I3)
  134 FORMAT(/" THE DATA ENTERED IS:"/
     .         5X," LAPSE RATE  5PPM  25PPM  SIGMA  RH"/
     .         12X,F4.1,2X,F4.1,3X,F4.1,3X,F4.1,1X,I3)
```

```
135 FORMAT(/" FOR TOWER NUMBER",1X,I3,1X,"THE DATA ENTERED IS:"/
     .         5X,"ALT  DIRECTION  SPEED  GUSTS"/)
136 FORMAT(" ENTER MODE: {&dJO=OPERATIONAL{&d@,P=PRODUCTION"
     .        ",R=RESEARCH _")
137 FORMAT(/" ENTER THE VEHICLE {&dJS=SHUTTLE{&d@,T=TITAN,D=DELTA _")
138 FORMAT(/" DO YOU WANT A DETAILED DATA FILE? (Y=YES,{&dJN=NO{&d@)"
     .        " _")
139 FORMAT( 5X,I3,8X,I3,5X,I2)
140 FORMAT(/" FOR TOWER NUMBER",1X,I3,1X,"THE DATA ENTERED IS:"/
     .         5X,"ALT  DIRECTION  SPEED  "/)
141 FORMAT(/" THE TEMPERATURES ENTERED ARE:"/5X,"ALTITUDE"
     .        " TEMPERATURE"/10X,I3,9X,I3/10X,I3,9X,I3)
142 FORMAT(/" IS THERE DATA FOR ANOTHER TOWER NUMBER USING THE SAME"
     .        " TIME? (Y=YES,{&dJN=NO{&d@) _")
143 FORMAT(/" ENTER DATA FOR WIND SYSTEM TOWER NUMBER",1X,I3)
200 FORMAT(3A2)
201 FORMAT(A1)
202 FORMAT(I5)
209 FORMAT(A2)
210 FORMAT(I2)
211 FORMAT(I4)
213 FORMAT(I3)
221 FORMAT(F4.1)
C
C SET THE VARIABLE 'MODE' IDENTIFYING THE RUNNING MODE; OPERATION, RE-
C SEARCH, OR PRODUCTION. IF IN OPERATION MODE, ENTER ONLY THE DATA
C THAT IS ABSOLUTELY NECESSARY...IF IN THE PRODUCTION MODE, ALLOW
C THE USER TO DECIDE WHETHER OR NOT TO ENTER ALL OR ONLY ESSENTIAL
C DATA...IF IN RESEARCH MODE, ENTER ALL DATA. THE VARIABLE, 'IQIK'
C TESTS FOR ALL OR ONLY ESSENTIAL DATA INPUT.
C
      IQIK = 1
      IF(IRUN.EQ.2) GO TO 1
      IQIK = 0
      IF(IRUN.EQ.1) GO TO 24
      WRITE (LU,138)
      READ (LU,201) IANS
      IF(IANS.NE.INY) IQIK = 1
      IF(IQIK.EQ.1) GO TO 1
C
C GET THE NAME OF THE DATA FILE TO BE USED OR CREATED
C --LET THE FILE NAMED "TOWERS" BE CREATED IF ONLY ESSENTIAL DATA
C   IS BEING ENTERED.
C
   24 WRITE (LU,100)
      READ (LU,200)NAM2
C
C OPEN OR CREATE THE DATA FILE NEEDED
C
    1 CALL OPEN(NDCB,IERR,NAM2,0)
      IF(IERR.LT.0) CALL CREAT(NDCB,IERR,NAM2,4,4,0,0,144)
      IF(IERR.LT.0) WRITE (LU,101)IERR
C
C WRITE THE TEST NUMBER TO THE FILE -- LET THE TEST NUMBER DEFAULT
C TO 00999 IF ENTERING ESSENTIAL DATA ONLY.
C
      IF(IQIK.EQ.1) NTEST = 00999
      IF(IQIK.EQ.1) GO TO 2
      WRITE (LU,102)
      READ (LU,202)NTEST
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
    2 CALL CODE
      WRITE(NBUF,104)LINE1,NTEST
      CALL CODE
      CALL WRITF(NDCB,IERR,NBUF,7)
      IF(IERR.LT.0) WRITE(LU,101)IERR
C
C WRITE WIND SYSTEM TOWER DATA HEADING
C
      CALL CODE
      WRITE (NBUF,103)LINE2
      CALL CODE
      CALL WRITF(NDCB,IERR,NBUF,11)
      IF(IERR.LT.0) WRITE (LU,101) IERR
C
C WRITE THE CAPE CANAVERAL HEADING
C
      CALL CODE
      WRITE (NBUF,105)LINE3
      CALL CODE
      CALL WRITF(NDCB,IERR,NBUF,12)
      IF(IERR.LT.0) WRITE(LU,101) IERR
C
C WRITE THE FIRST LINE OF THE HEADER FOR THE DATA COLUMNS
C
      CALL CODE
      WRITE (NBUF,106)LINE4
      CALL CODE
      CALL WRITF(NDCB,IERR,NBUF,50)
      IF(IERR.LT.0) WRITE (LU,101) IERR
C
C WRITE THE SECOND LINE OF THE DATA COLUMNS HEADINGS
C
      CALL CODE
      WRITE (NBUF,106)LINE5
      CALL CODE
      CALL WRITF(NDCB,IERR,NBUF,50)
      IF(IERR.LT.0) WRITE (LU,101) IERR
C
C WRITE THE THIRD LINE OF THE HEADINGS FOR THE DATA COLUMNS
C
      CALL CODE
      WRITE (NBUF,108)LINE6
      CALL CODE
      CALL WRITF(NDCB,IERR,NBUF,34)
      IF(IERR.LT.0) WRITE(LU,101) IERR
C
C ENTER THE DATE, TIME, AND TIME INTERVAL.  LET THE TIME AND DATE
C DEFAULT TO 99 AND 9999 IF ONLY ESSENTIAL DATA IS BEING ENTERED.
C
      IF(IQIK.EQ.1) GO TO 4
    3 WRITE (LU,110)
      READ (LU,210) IYR
      WRITE (LU,111)
      READ (LU,210)MON
      WRITE (LU,112)
      READ (LU,210) IDAY
      WRITE (LU,113)
      READ (LU,211) ITIM
      WRITE (LU,114)
      READ (LU,210) INT
```

```
      WRITE (LU,131) IYR,MON,IDAY,ITIM,INT
      WRITE (LU,109)
      READ (LU,269) IANS
      IF(IANS.EQ.INN) GO TO  3
      IF(IANS.EQ INS)GO TO 999
      GO TO 5
    4 IYR = 99
      MON = 99
      IDAY = 99
      ITIM = 9999
      INT = 99
C
C SET THE TOWER NUMBER ACCORDING TO THE LAUNCH VEHICLE; IF SHUTTLE,
C TOWER NUMBER = 313, IF TITAN, TOWER NUMBER = 110, IF DELTA, TOWER
C NUMBER = 303.
C
C IF IN THE RESEARCH MODE, ENTER THE TOWER NUMBER
C
    5 IF(IRUN.EQ.1) GO TO 23
      IF(IVHICL.EQ.0) NTWR = 313
      IF(IVHICL.EQ.1) NTWR = 110
      IF(IVHICL.EQ.2) NTWR = 303
      GO TO 6
   23 WRITE (LU,115)
      READ (LU,213) NTWR
C
C IF NOT ENTERING ONLY ESSENTIAL DATA, ASK FOR INPUT AT EACH LEVEL.
C
C IF ENTERING ESSENTIAL DATA ONLY, ASK FOR INPUT AT THE 12, 54, AND TOP
C OF THE TOWER LEVELS
C
    6 IF(IQIK.EQ.1) GO TO 8
      DO 7 K=1,7
      WRITE (LU,130) LEVEL(K)
      WRITE (LU,116)
      READ (LU,213) IWD(K)
      WRITE (LU,117)
      READ (LU,210) IWS(K)
      WRITE (LU,118)
      READ (LU,210) IGS(K)
    7 CONTINUE
      GO TO 16
C
C IF ENTERING ESSENTIAL DATA ONLY, BEGIN BY GETTING DATA FOR THE
C 12 AND 54 FOOT LEVEL REGARDLESS OF THE LAUNCH VEHICLE.
C SET THE WIND GUST SPEEDS TO 99.
C
    8 WRITE (LU,143) NTWR
      DO 9 J=1,2
      WRITE (LU,130)LEVEL(J)
      WRITE (LU,116)
      READ (LU,213) IWD(J)
      WRITE (LU,117)
      READ (LU,210) IWS(J)
      IGS(J) = 99
    9 CONTINUE
C
C IF ENTERING ONLY ESSENTIAL DATA SET ALL DATA AT 162, 204, 295,
C 394, AND 492 FOOT LEVELS TO 99 OR 999.
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 11 J=3,7
      IWD(J) = 999
      IWS(J) = 99
      IGS(J) = 99
   11 CONTINUE
C
C IF ENTERING ESSENTIAL DATA ONLY AND IF THE LAUNCH VEHICLE IS A
C SHUTTLE CRAFT, THEN PICK UP THE TOP OF THE TOWER DATA (IE;492 FT.)
C
C IF THE LAUNCH VEHICLE IS TITAN, THEN PICK UP THE TOP OF THE TOWER
C DATA (IE; 204 FOOT LEVEL)
C
      IF(IVHICL.EQ.0) GO TO 15
      IF(IVHICL.EQ.1) GO TO 13
      GO TO 16
   13 WRITE (LU,130) LEVEL(4)
      WRITE (LU,116)
      READ (LU,213) IWD(4)
      WRITE (LU,117)
      READ (LU,210) IWS(4)
      GO TO 16
   15 WRITE (LU,130) LEVEL(7)
      WRITE (LU,116)
      READ (LU,213) IWD(7)
      WRITE (LU,117)
      READ (LU,210) IWS(7)
C
C WRITE THE VALUES ENTERED BACK TO THE CRT FOR USER APPROVAL BEFORE
C CONTINUING
C
   16 IF(IQIK.EQ.0) WRITE (LU,135)HTWR
      IF(IQIK.EQ.1) WRITE (LU,140)HTWR
      DO 17 K=1,7
      IF(IQIK.EQ.1.AND.IWD(K).EQ.999) GO TO 17
      IF(IQIK.EQ.1) WRITE (LU,139) LEVEL(K),IWD(K),IWS(K)
      IF(IQIK.EQ.0) WRITE (LU,132)LEVEL(K),IWD(X),IWS(K),IGS(K)
   17 CONTINUE
      WRITE (LU,109)
      READ (LU,209) IANS
      IF(IANS.EQ.1HN) GO TO 6
      IF(IANS.EQ.1HS) GO TO 999
C
C ENTER THE TEMPERATURE AND DEW POINT FOR THE 6 FOOT AND 54 FOOT
C LEVELS IF ENTERING ALL DATA, OTHERWISE JUST ENTER THE TEMPERATURES.
C
   18 DO 19 K=1,2
      WRITE (LU,119) LEVL2(K)
      READ (LU,213) ITEM(K)
      IF(IQIK.EQ.1) IDP(K) = 999
      IF(IQIK.EQ.1) GO TO 19
      WRITE (LU,120) LEVL2(K)
      READ (LU,213) IDP(K)
   19 CONTINUE
      IF(IQIK.EQ.1) WRITE (LU,141) LEVL2(1),ITEM(1),LEVL2(2),ITEM(2)
      IF(IQIK.EQ.0) WRITE (LU,133)LEVL2(1),ITEM(1),IDP(1),LEVL2(2),
                                        ITEM(2),IDP(2)
      WRITE (LU,109)
      READ (LU,209) IANS
      IF(IANS.EQ.1HN) GO TO 18
      IF(IANS.EQ.1HS) GO TO 999
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C IF ENTERING ONLY ESSENTIAL DATA, SET LAPSE RATE,
C 5 PPM DISTANCE, 25 PPM DISTANCE, SIGMA OF WIND AZIMUTH, AND
C RELATIVE HUMIDITY TO 99.9 OR 99.
C
C ENTER THE LAPSE RATE, 5 PPM DISTANCE, 25 PPM DISTANCE, THE STANDARD
C DEVIATION OF THE WIND AZIMUTH ANGLE, AND THE RELATIVE HUMIDITY AT
C 54 FEET IF ENTERING ALL DATA.
C
      IF(IQIK.EQ.1) GO TO 21
   20 WRITE (LU,121)
      READ (LU,221) RATEL
      WRITE (LU,122)
      READ (LU,221) PPM5
      WRITE (LU,123)
      READ (LU,221) PPM25
      WRITE (LU,124)
      READ (LU,221) SDEV
      WRITE (LU,125)
      READ (LU,213) IRH
      WRITE (LU,134)RATEL,PPM5,PPM25,SDEV,IRH
      WRITE (LU,109)
      READ (LU,209) IANS
      IF(IANS.EQ.IHN) GO TO 20
      IF(IANS.EQ.IHS) GO TO 999
      GO TO 22
   21 RATEL = 99.9
      PPM5 = 99.9
      PPM25 = 99.9
      SDEV = 99.9
      IRH = 999
C
C WRITE THE DATA TO THE DISC FILE
C
   22 CALL CODE
      WRITE(NBUF,126) IYR,MON,IDAY,ITIM,INT,NTWR,IWD(1),IWS(1),IGS(1),
     .                IWD(2),IWS(2),IGS(2),IWD(3),IWS(3),IGS(3),IWD(4),
     .                IWS(4),IGS(4),ITEM(1),IDP(1),IDP(2),RATEL,PPM5,
     .                PPM25,SDEV,ITEM(2),IRH
      CALL CODE
      CALL WRITF(NDCB,IERR,NBUF,58)
      IF(IERR.LT.0)WRITE (LU,101) IERR
      CALL CODE
      WRITE (NBUF,120) IWD(5),IWS(5),IGS(5),IWD(6),IWS(6),IGS(6),IWD(7),
     .                 IWS(7),IGS(7)
      CALL CODE
      CALL WRITF(NDCB,IERR,NBUF,58)
C
C SEE IF THERE IS DATA FOR ANOTHER TOWER USING THE SAME DATE AND TIME
C IF ENTERING ALL DATA.--IF IN OPERATIONAL MODE, DO NOT ASK FOR OTHER
C LAUNCH VEHICLES--AND CLOSE THE FILE. IF IN OPERATIONAL MODE, TEST
C FOR ESSENIAL DATA ONLY OR ALL DATA INPUT.
C
C IF OTHER LAUNCH VEHICLES OR TOWER NUMBERS ARE TO BE INPUT IN THE
C PRODUCTION OR RESEARCH MODE, LOOP BACK TO THE BEGINNING OF THE
C PROGRAM.
C
      IF(IRUN.EQ.2) GO TO 999
      IF(IRUN.EQ.3) WRITE (LU,129)
      IF(IRUN.EQ.1) WRITE (LU,142)
```

```
      READ (LU,209) IANS
      IF(IANS.EQ.1HY) GO TO 5
C
C CLOSE THE DATA FILE BEING USED
C
  999 NBUF(1) = 2HEN
      CALL CODE
      CALL WRITF(NDCB,IERR,NBUF,1)
      IF(IERR.LT.0) WRITE (LU,101)IERR
      CALL CLOSE(NDCB,IERR)
      IF(IERR.LT.0) WRITE (LU,101) IERR
C
C TELL THE USER HIS DATA FILE NAME AGAIN
C
      WRITE (LU,127) NAM2
C
C
      DO 25 K=1,3
 25   NDAT(K) = NAM2(K)
      NDAT = NAM2
C
C TERMINATE TOWER
C
      RETURN
      END
      SUBROUTINE FORTD(NAM2,IALT,KNUM)
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C++                                                      ++
C++ THIS PROGRAM READS WIND SYSTEM TOWER DATA IN TMX-    ++
C++ 73337 FORMAT AND REFORMATS THE DATA TO CONFORM TO THE++
C++ SOUNDING/FORECAST DATA FORMAT.                       ++
C++                                                      ++
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C
C  >>>   COMMON AREAS
C
C
C  >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGMCL
C
C  >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNOFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,NONLY,MFORM,LUFRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C  >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C  >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
```

```
C  >>>   LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNID(10)
      LOGICAL LTIME, LVERSN
C
C  >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C  >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGA7
C
C  >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C  >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
C DIMENSION AND DATA STATEMENTS
C
      DIMENSION NAM2(3),NDCB(144),NBUF(58),IWD(30),IWS(30),
     .          ITEM(15),LEVEL(7),NAM3(3),KDCB(144),KBUF(58)
      DIMENSION IALT(30),DPTEMP(30),NAME(3)
      DATA LEVEL/12,54,162,204,295,394,492/
      DATA NAME/2H=R,2HEE,2HD2/
      DATA NAM3/2HRE,2HAD,2HER/
C
C FORMAT STATEMENTS
C
  100 FORMAT("OPEN ERROR",I4)
  101 FORMAT(" IERR =",I4)
  102 FORMAT("CLOSE ERROR",I4)
  106 FORMAT(58A2)
  112 FORMAT( "KERR =",I3)
  144 FORMAT(36X,3(I3,2X,I2,5X,))
  145 FORMAT("TWR NBR",1X,I3,1X)
  146 FORMAT(I6,1XI3,1XI3,F6.1)
  147 FORMAT(I6,1XI3,1XI3)
  203 FORMAT(A2)
  214 FORMAT(20X,I3,1X,4(I3,2X,I2,5X),I3,32X,I3)
  215 FORMAT(I6,1XI3,1XF4.1,F6.1)
  216 FORMAT(I6,1XI3,1XF4.1)
  248 FORMAT(3A2)
C
C SET THE TOWER NUMBER ACCORDING TO THE VEHICLE NUMBER PASSED
C TO THE SUBROUTINE
C
      NTWR = 313
      IF (IVHICL.EQ.1) NTWR = 110
      IF (IVHICL.EQ.2) NTWR = 303
C
C OPEN THE FILE SPECIFIED
C
      CALL OPEN (NDCB,IERR,NAM2,0)
      IF(IERR.LT.0) WRITE (LU,100) IERR
```

```
C
C CREATE THE DISC FILE "READER"
C
      CALL OPEN(KDCB,KERR,NAM3,0)
      IF(KERR.LT.0) CALL CREAT(KDCB,KERR,NAM3,4,4,0,0,144)
      IF (KERR.LT.0) WRITE (LU,101) KERR
C
C READ LINES OF THE DATA FILE UNTIL DATA IS FOUND
C
    3 CALL READF(NDCB,IERR,NBUF,50,LEN)
      IF(IERR.LT.0) WRITE (LU,101) IERR
      IF(NBUF(1).EQ.2HYR) GO TO 4
      IF(NBUF(1).EQ.2HEN) GO TO 99
      GO TO 3
C
C READ THE DATA HEADINGS
C
    4 CONTINUE
      CALL CODE
      READ (NBUF,106) (NBUF(I),I=1,LEN)
      CALL READF(NDCB,IERR,NBUF,50,LEN)
      IF(IERR.LT.0) WRITE (LU,101) IERR
      IF(NBUF(1).EQ.2HEN) GO TO 99
      CALL CODE
      READ (NBUF,106) (NBUF(I),I=1,LEN)
      CALL CODE
      CALL READF(NDCB,IERR,NBUF,50,LEN)
      IF(IERR.LT.0) WRITE (LU,101) IERR
      IF(NBUF(1).EQ.2HEN) GO TO 99
      CALL CODE
      READ (NBUF,106) (NBUF(I),I=1,LEN)
C
C READ THE DATA FROM THE TMX 73337 FORMATTED FILE AND WRITE IT
C TO THE DISC FILE "READER" IN THE SAME FORMAT AS THE SOUNDING/
C FORECAST
C
      DO 7 K=1,22
      CALL READF(NDCB,IERR,NBUF,50,LEN)
      IF(IERR.LT.0) WRITE (LU,101) IERR
      IF(NBUF(1).EQ.2HEN) GO TO 99
      CALL CODE
      READ (NBUF,214)ITUR,IWD(1),IWS(1),IWD(2),IWS(2),IWD(3),IWS(3)
     ,           ,IWD(4),IWS(4),ITEM(1),ITEM(2)
      CALL CODE
      IF(ITUR.NE.NTUR) GO TO 7
      CALL READF(NDCB,IERR,NBUF,50,LEN)
      IF(NBUF(1).EQ.2HEN) GO TO 99
      IF(IERR.LT.0) WRITE (LU,101) IERR
      CALL CODE
      READ (NBUF,144) IWD(5),IWS(5),IWD(6),IWS(6),IWD(7),IWS(7)
      CALL CODE
      WRITE (KBUF,145) ITUR
      CALL CODE
      CALL WRITF(KDCB,KERR,KBUF,6)
      CALL CODE
      WRITE (KBUF,146) LEVEL(1),IWD(1),IWS(1),ITEM(1)
      CALL CODE
      CALL WRITF(KDCB,KERR,KBUF,10)
      IF(KERR.LT.0) WRITE (LU,101) KERR
        .I C(..
```

```
      WRITE (KBUF,146) LEVEL(2),IWD(2),IWS(2),ITEM(2)
      CALL CODE
      CALL WRITF(KDCB,KERR,KBUF,10)
      IF(KERR.LT.0) WRITE (LU,101) KERR
      IF(ITWR.EQ.303) GO TO 99
      I = 7
      IF(ITWR.EQ.110) I = 4
      CALL CODE
      WRITE (KBUF,147) LEVEL(I),IWD(I),IWS(I)
      CALL CODE
      CALL WRITF(KDCB,KERR,KBUF,?)
      IF(KERR.LT.0) WRITE (LU,101) KERR
    7 CONTINUE
C
C PUT AN END OF FILE MARK ON THE DATA FILE "READER" AND
C CLOSE BOTH FILES.
C
   99 NBUF(1) = 2HEN
      CALL CODE
      WRITE (KBUF,203) NBUF(1)
      CALL CODE
      CALL WRITF(KDCB,KERR,KBUF,1)
      CALL CLOSE(NDCB,IERR)
      CALL CLOSE (KDCB,KERR)
      IF(IERR.LT.0) WRITE (LU,102) IERR
      IF(KERR.LT.0) WRITE (LU,102) KERR
      CALL OPEN(KDCB,KERR,NAM3,0)
      IF(KERR.LT.0) WRITE (LU,112) KERR
C
C LOOP UNTIL THE CORRECT TOWER NUMBER IS FOUND
C
   33 CALL READF(KDCB,KERR,KBUF,40,LEN)
      IF (KERR.LT.0) WRITE (LU,112) KERR
      IF (KBUF(1).EQ.2HEN) GO TO 999
C
C READ IN THE DATA
C
      NUM1 = NUM + 1
      NUM2 = NUM + 2
      NUM3 = NUM + 3
      DO 5 J=NUM1,NUM2
      CALL READF(KDCB,KERR,KBUF,40,LEN)
      IF (KERR.LT.0) WRITE (LU,112) KERR
      IF (KBUF(1).EQ.2HEN) GO TO 999
      CALL CODE
      READ (KBUF,215) IALT(J),IDIR(J),SPEED(J),TEMP(J)
C
C CONVERT THE TEMPERATURE IN THE TOWER DATA FROM DEGREES
C F TO DEGREES C.
C
      TEMP(J) = .555555*(TEMP(J)-32.0)
    5 CONTINUE
C
C IF THE LAUNCH VEHICLE IS A DELTA (TOWER NUMBER 303),
C THERE IS NO OTHER DATA
C
      IF(IVHICL.EQ.2) GO TO 999
      CALL READF(KDCB,KERR,KBUF,40,LEN)
      IF (KERR.LT.0) WRITE (LU,101) KERR
      IF (KBUF(1).EQ.2HEN) GO TO 999
```

```
      CALL CODE
      READ (KBUF,216) IALT(NUM3),IDIR(NUM3),SPEED(NUM3)
C
C CLOSE THE DATA FILE "READER"
C
  995 CALL CLOSE(KDCB,KERR)
      IF (KERR.LT.0) WRITE (LU,101) KERR
      IF(IVHICL.EQ.2) KNUM = NUM + 2
      IF(IVHICL.NE.2) KNUM = NUM + 3
      DO 95 J=NUM1,KNUM
      DPTEMP(J) = 0.0
   95 PRESS(J) = 0.0
C
C TERMINATE FORTD
C
      RETURN
      END
      END$
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
&RCLDS T=00004 IS ON CR00003 USING 00071 BLKS R=0000

FTN,L
      PROGRAM RCLDS(3),REEDS CLOUD RISE PROGRAM              *CAB,810814
C
C     ***************************************************************
C     *                                                             *
C     *    CLOUD RISE PROGRAM -- A PROGRAM IN THE REED SERIES OF    *
C     *    PROGRAMS                                                 *
C     *                                                             *
C     ***************************************************************
C
C
C  >>>   COMMON AREAS
C
C
C  >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C  >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNDFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,MONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C  >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C  >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMOIK 2),ISYEAR
C
C  >>>   LAUNCH TIME
      COMMON LTIME,LTIN.LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME, LVERSN
C
C  >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIN,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C  >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C  >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C  >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
                  (HEATM,VPAR(10)),(HTOT,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),[illegible]VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PHO,VPAR(17)),(GAMMAX,VPAR(18))
```

```
C
C*** INPUT FORMATS
90    FORMAT(A1)
C
C        OUTPUT FORMAT STATEMENTS
C
  100 FORMAT (F6.1)
  101 FORMAT (F7.2)
  102 FORMAT (F4.1)
  200 FORMAT ("1"27X"EXHAUST CLOUD"/"0LEVEL"4X"ALTITUDE"17X
     .         "RISE TIME"5X"RANGE"6X"DIRECTION"/10X"(METERS)"17X
     .         "(SECONDS)"4X"(METERS)"4X"(DEGREES)")
  201 FORMAT (2XI3,5XF7.1,5X"ADIABATIC"5X3A2,6XF7.1,7XF5.1)
  202 FORMAT (2XI3,5XF7.1,6X"STABLE"7X3A2,6XF7.1,7XF5.1)
  203 FORMAT (//"0****CLOUD STABLIZATION****"/
     .         6X"HEIGHT(M): "F6.1/
     .         6X"STABILIZATION TIME AFTER LAUNCH(SEC): "F5.1/
     .         6X"RANGE FROM PAD(M): "F7.1/
     .         6X"DIRECTION FROM PAD(DEG): "F5.1)
  204 FORMAT (F6.1)
  205 FORMAT (//"0****TOP OF SURFACE LAYER METEOROLOG CAL PARAMETERS"
     .         "****"/
     .         6X"HEIGHT(M): "F6.1/
     .         6X"WIND DIRECTION(DEG): "I3/
     .         6X"WIND SPEED(M/SEC): "F4.1)
  206 FORMAT (//"0****DIFFUSION PARAMETERS****"/
     .         6X"MEAN SPEED(M/SEC): "F4.1/
     .         6X"MEAN TRANSPORT DIRECTION(DEG): "F5.1)
  207 FORMAT (F3.0)
  208 FORMAT (//"0SIGMA OF WIND AZIMUTH ANGLE, SIGAZ: "F4.1)
  209 FORMAT (//"0EFFECTIVE CLOUD HEIGHT(M): "F6.1)
C
C        TYPE AND DIMENSION STATEMENTS
C
      INTEGER RMETP(3),RCONC(3)
      REAL LEASTD
      DIMENSION NAME(3),NAMEF(3),ILINE(32),IDATAF(10),IERS(32)
C
C        EQUIVALENCE STATEMENT
C
      EQUIVALENCE (ILIN23,ILINE(23)),(ILIN24,ILINE(24))
C
C        DATA STATEMENTS
C
      DATA NAME/2H=R,2HEE,2HD2/, NAMEF/2H?R,2HCL,2HDS/
      DATA RMETP/2HRM,2HTP,1HS/
      DATA RCONC/2HRC,2HON,1HS/
      DATA IERS/32*2H  /
C
C        CALL SUBROUTINE GRAF TO INITIALIZE PLASMASCOPE GRAPHIC MODE
C
      CALL GRAF(1)
      CALL DREAD(NAMEF,2,ILINE)
      CALL DREAD(NAMEF,2,ILINE)
C
C        READ THE COMMON DATA FILE
C
      CALL RWDIS(NAME,0)
C*** 2ND CALL TO DREAD TO TEST OPEN
      [illegible]
```

```
C
C         INITIALIZE SOME LOCAL VARIABLES
C
C         RISTIM( ) - CLOUD RISE TIME
C         IAS - 0 = ADIABATIC
C               1 = STABLE
C         ALTINC - ALTITUDE INCREMENT
C         ITERAT - ITERATION COUNTER
C
      RNGY = 0.0
      RNGX = 0.0
      RISTIM(1) = 0.0
      ALTINC = 0.0
      STBRNG = 0.0
      STBAZD = 0.0
C
C*** INITIALIZE DISPLAY LU FLAG
      CALL VERSN(I)
      LVERSN = I .EQ. 0
C
C         WRITE OUT THE EXHAUST CLOUD HEADER
C
      WRITE (LUPRNT,200)
C
C         CALCULATE SOME QUANTITIES TO BE USED IN SUBSEQUENT DO LOOP
C
      ALPHAC = 5.12913006E-02*(TEMP(1) + 273.16)*SURDEN*GAMMAX**3
      ALPHAC = ALPHAC/(HEATN * QC1)
      GT = 9.8/(TEMP(1) + 273.16)
C
C         DO LOOP TO CALCULATE EXHAUST CLOUD PARAMETERS
C
CC***THE FOLLOWING CHANGES ARE AN ATTEMPT TO USE THE ALGORITHM
CC***WRITTEN IN CR-2631, PAGE 15, FOR A LEAST SQUARE REGRESSION
CC***FIT FOR THE HEIGHT RATE OF CHANGE FOR POTENTIAL TEMPERATURE.
CC***...JSH
      IFLAG = 1
      ZSUM = 0.0
      PTSUM = 0.0
      S1 = 0.0
      S2 = 0.0
C
      DO 9 I=2,NUM
      IM1 = I - 1
      IAS = 1
      POT = PTEMP(I) -273.16
C
C         CALCULATE SLOPE OF POTENTIAL TEMPERATURE, SPEED, AND
C         DIRECTION IN LAYER
C
      DALT = ALT(I) - ALT(IM1)
      GPTEMP = (PTEMP(I) - PTEMP(IM1))/DALT
      GSPEED = (SPEED(I) - SPEED(IM1))/DALT
      DDEG = ABS(FLOAT(IDIR(I) - IDIR(IM1)))
      IF(DDEG.GT.180.) DDEG = 360.0 - DDEG
      GDIR = DDEG/DALT
      IF(IPTZ.EQ.1)GO TO 57
C
C         CALCULATE METEOROLOGICAL [illegible]
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
    1 IF(ALTINC.EQ.0.0)GO TO 44
      S1 = S1 - S1INC
      S2 = S2 - S2INC
      ZSUM = ZSUM - Z
      PTSUM = PTSUM - PT
      Z = ALT(I) - ALTINC
      TERP = (ALT(I) - ALTINC - ALT(IM1))/ DALT
      PT = POT*TERP
      GO TO 55
   44 Z = ALT(I)
      PT = POT
   55 ZSUM = ZSUM + Z
      PTSUM = PTSUM + PT
      ZAV = ZSUM/FLOAT(I)
      PTAV = PTSUM/FLOAT(I)
      S1INC = (Z - ZAV)*(PT - PTAV)
      S2INC = (Z - ZAV)**2
      S1 = S1 + S1INC
      S2 = S2 + S2INC
      STAB = GT*(S1/S2)
CC    WRITE(6,1232)I,STAB,ALTINC,Z,S1,PT,PTAV
 1232 FORMAT(1X,"I,STAB,ALTINC,Z,S1,PT,PTAV=",1X,I3,6(1X,E10.8))
   57 ZALPHA = ALT(I) - ALT(1) - ALTINC
      ALPHA = ALPHAC + ZALPHA**4/(AA + ZALPHA**BB + CC)
C
C        CALCULATE POTENTIAL TEMPERATURE FACTOR
C
      IF(IPTZ.EQ.1)STAB = GT + (PTEMP(I)- ALTINC + GPTEMP - PTEMP(1))/
     *            (ALT(I) - ALTINC - ALT(1) + 1.0E-7)
C
C        CALCULATION FOR ADIABATIC RISE
C
      IF(STAB .GT. 0.000001)GO TO 2
      RISTIM(I) = SQRT(ALPHA)
      IAS = 0
      GO TO 5
C
C        CALCULATION FOR STABLE CLOUD RISE
C
C        C2 - ARGUMENT OF ARC COSINE (MUST BE LESS THAN -1)
C
    2 C2 = 1.0 - 0.5 * ALPHA * STAB
CC    WRITE(6,122)I,ALPHA,STAB,C2
  122 FORMAT(1X,"I,ALPHA,STAB,C2 =",1X,I3,3(1X,E14.8))
      IF(C2 .LT. -1.0)GO TO 4
      C3 = C2/SQRT(1.0 - C2 * C2)
      RISTIM(I) = (PIOVR2 - ATAN(C3))/SQRT(STAB)
      IF(IFLAG .EQ.0) GO TO 11
      GO TO 5
C
C        ITERATE IN LAYER
C
    4 ALTINC = ALTINC + 0.1
      IFLAG = 0
      IF(IPTZ.EQ.1)GO TO 57
      GO TO 1
C
C        CALCULATE RANGE AND DIRECTION
C
    5    HG =    1.5    FEET ...
```

```
     .         (RISTIM(IMI) - RISTIM(I))
C
C*** AVERAGE WIND DIRECTION...BE CAREFUL TO ADJUST FOR PROPER VECTOR DIRECTION
C
      DELDIR = DEGRAD * (.5*(FLOAT(IDIR(I) + IDIR(IMI))) -
     .         180.*(IABS(IDIR(I) - IDIR(IMI))/180))
      DIRECT = RADDEG * DELDIR
C
C*** FIND ACTUAL WIND TRANSPORT DIRCETION..ADD PI FOR DELDIR <180..ELSE SUB
C
      IFLIP = 1
      IF(DELDIR .GT. PI) IFLIP = -1
      DELDIR = DELDIR + IFLIP*PI
      RNGY = RNGY + STBRNG * SIN(PI - DELDIR)
      RNGX = RNGX + STBRNG * COS(PI - DELDIR)
      STBAZD =  ATAN2(RNGY,RNGX)
      STBAZD = RADDEG * (PI - STBAZD)
      STBRNG = SQRT(RNGY * RNGY + RNGX * RNGX)
C
C        WRITE OUT THE VARIABLES WITH THE APPROPRIATE FORMAT STATEMENT
C        BASED OF WHETHER OR NOT CLOUD IS ADIABATIC OR STABLE
C
      CALL BLANK(IDATAF,3)
      IF(ALTINC .GT. 0.0)GO TO 6
      CALL CODE
      WRITE (IDATAF,100) RISTIM(I)
    6 IF(IAS .NE. 0)GO TO 7
      WRITE (LUPRNT,201) I,ALT(I),(IDATAF(J),J=1,3),STBRNG,STBAZD
      GO TO 8
    7 WRITE (LUPRNT,202) I,ALT(I),(IDATAF(J),J=1,3),STBRNG,STBAZD
    8 IF(ALTINC .NE. 0.0)GO TO 11
    9 CONTINUE
C
C        CALCULATE AND WRITE OUT STABILIZATION HEIGHT AND TIME
C
   11 STBRNG = 0.5 * (SPEED(IMI) - ALTINC * GSPEED + SPEED(I)) *
     .           (RISTIM(I) - RISTIM(IMI))
      DALT = 0.00872665 * (FLOAT(IDIR(I) + IDIR(IMI)) - GDIR * ALTINC)
      RNGY = RNGY - STBRNG * SIN(PI - DALT)
      RNGX = RNGX - STBRNG * COS(PI - DALT)
      STBAZD =  ATAN2(RNGY,RNGX)
      STBAZD = RADDEG * (PI - STBAZD)
      STBRNG = SQRT(RNGY * RNGY + RNGX * RNGX)
      STBALT = ALT(I) - ALTINC
      WRITE (LUPRNT,203) STBALT,RISTIM(I),STBRNG,STBAZD
      TIST = PI/(SQRT(STAB))
C
C        STORE THE INDEX OF THE ESTIMATED TOP OF THE SURFACE LAYER
C
      LAYTOP = I
C
C        LOAD THE CLOUD RISE TIME ARRAY
C
      STBTIM = RISTIM(LAYTOP)
      I = I + 1
      DO 12 J=I,INUM
   12 RISTIM(J) = STBTIM
C
C        CALL FTOP TO COMPUTE THE INDEX OF THE TOP OF THE SURFACE LAYER
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CALL FTOP
C
C        IS THIS A RESEARCH, OPERATIONAL, OR PRODUCTION RUN?
C
      IF(IRUN .LE. 2)GO TO 18
C
C        PRODUCTION RUN -- IF TOPLAY IS UNDEFINED, USE LAYTOP AS ESTIMATED
C
   15 IF(TOPLAY .LE. 0.0)GO TO 26
C
C        CALCULATE LAYTOP BASED ON VALUE OF TOPLAY
C
      LEASTD = 9999999.9
      DO 16 I=1,NUM
      DIFF = ABS(ALT(I) - TOPLAY)
      IF(DIFF .GT. LEASTD)GO TO 16
      LEASTD = DIFF
      LAYTOP = I
   16 CONTINUE
      GO TO 26
C
C        CALL THE SUBROUTINE RSGAZ TO CALCULATE THE VALUE OF SIGAZ
C        NOTE THAT THE FLAG ITDU IS USED TO BRANCH WHEN TOWER DATA
C        IS BEING USED.
C
   18 J1 = 1
      J2 = 0
      J3 = 0
      IF(ITDU.EQ.0) GO TO 17
      J1 = 1
      J2 = 3
      J3 = 4
      GO TO 25
   17 DO 19 J=1,NUM
C     IF(ABS(ALT(J)-304.8) .LE. 1.0)J3 = J
CC
CC ***THE TEST FOR DETERMINING J3 AND J2 HAS BEEN ALTERED SINCE 1.0
CC ***AND 1000.0 ARE TOO RIGID FOR WORKABLE STANDARDS.
CC
C     IF(ABS(ALT(J)-304.8) .LT. ABS(ALT(J3)-304.8))J3 = J
      IF(ALT(J).LE.304.8 .AND. ALT(J+1).GT.304.8) J3 = J
      IF(ABS(PRESS(J)-1000.0).LT. ABS(PRESS(J2)-1000.0))J2 = J
   19 CONTINUE
      SIGAZ = 7.0
   25 IF(J2.NE.0 .AND. J3.NE.0)CALL RSGAZ(J1,J2,J3,SIGAZ)
CC    WRITE (6,787) SIGAZ,NAMEF,LAYTOP
  787 FORMAT(" RSGAZ IN RCLDB IS:",F8.4,"   FILE:",3A2," LAYTOP:",I4)
C
C        WRITE OUT THE ESTIMATED TOP OF SURFACE LAYER -- READ IN
C        THE ONE TO BE USED -- CALCULATE LAYTOP
C
      CALL DREAD(NAMEF,2,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
      CALL CODE
      WRITE (IDATAF,204) ALT(LAYTOP)
      TOPLAY = ALT(LAYTOP)
      CALL CHAR(320.0,YPOS,0,IDATAF,6,0,0)
      YPOS = YPOS - 32.0
```

```
      IF(IRUN .GE. 2)GO TO 24
      CALL DREAD(NAMEF,3,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,6,3,0)
      CALL CHAR(56.0,YPOS,0,IERS,1,3,0)
      CALL CHAR(64.0,YPOS,0,ILINE(5),9,3,0)
      CALL CHAR(160.0,YPOS,0,ILINE(11),44,0,0)
      NIN = 6
      CALL BLANK(IDATAF,3)
      IX = 0
      TOPIX = IX
      IF(.NOT.LVERSN) GO TO 20
      CALL IN(2,JTYPE,463.0,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      IF(JTYPE .EQ. 1)GO TO 22
      GO TO 21
20    CONTINUE
      READ(LU,101)TOPIN
      IF(TOPIN.EQ.1H0)STOP
      IF(TOPIN .LE. TOPIX) GO TO 22
      TOPLAY = TOPIN
      IX = 20
      CALL CHAR(0.0,YPOS,0,ILINE,10,0,0)
      CALL CHAR(47.0,YPOS,0,IERS,40,0,0)
      YPOS = YPOS - 32
      GO TO 22
21    CONTINUE
      CALL CHAR(0.0,YPOS,0,ILINE,6,0,0)
      CALL CHAR(47.0,YPOS,0,IERS,40,0,0)
      YPOS = YPOS - 32.0
      CALL CODE
      READ (IDATAF,101) TOPLAY
      GO TO 15
   22 IF(IX .GT. 9)GO TO 23
      CALL CHAR(0.0,YPOS,0,ILINE,10,0,0)
      CALL CHAR(143.0,YPOS,0,IERS,46,0,0)
      YPOS = YPOS - 32.0
      GO TO 15
   23 CALL CHAR(0.0,YPOS,0,ILINE,6,0,0)
      CALL CHAR(56.0,YPOS,0,IERS,10,0,0)
      CALL CHAR(360.0,YPOS,0,IERS,10,0,0)
      YPOS = YPOS - 32.0
C
C        CALL MET PROFILE ROUTINE RMETP TO DETERMINE LAYER VALUE
C
   24 CALL NGRAF
      CALL RUDIS(NAME,1)
CBJ
CBJ   WRITE(6,700)
700   FORMAT("    **** CALLING RMTPS ****")
CBJ
      CALL EXEC(9,RMETP)
      CALL RUDIS(NAME,0)
C
C        IF MET PLOTS ARE ONLY BEING DONE, TERMINATE EXECUTION OF RCLOR
C
C*** WRITE TO THE PUNCH  IS TEMPORARILY DISABLED
C     WRITE(14,1414) SIGAZ,TOPLAY,ISTIME,ISDAY,ISMON(1),ISMON(2),
C    .               ISYEAR,SIGO
1414  FORMAT(F5.1,1X,F6.1,2X,I4,"Z",2X,I2,1X,2[illegible]
C***  IF(MONLY .EQ. 0)STOP
```

```
C
C         CHANGE THE TOPLAY VALUE?
C
      CALL GRAF(1)
CC    YPOS = 474.0
      IF(LVERSN) YPOS = 474.0
      CALL DREAD(NAMEF,5,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,50,0,0)
      CALL CODE
      WRITE (IDATAF,101) TOPLAY
      CALL CHAR(400.0,YPOS,0,IDATAF,7,0,0)
      ALT(LAYTOP) = TOPLAY
      YPOS = YPOS - 32.0
      GO TO 15
C
C         WRITE OUT THE TOP OF THE SURFACE LAYER AND WIND DIRECTION
C         AND SPEED AT THE TOP
C
   26 CONTINUE
      WRITE (LUPRNT,205) TOPLAY,IDIR(LAYTOP),SPEED(LAYTOP)
C
C         CALCULATE SOURCE STRENGTH
C
      SIGHCL = 2.276E3 * PHCL * QC1 * A.. * (TEMP(1) + 273.16)/
     .         PRESS(1) * TOPLAY**BB
C
C         CALCULATE AND WRITE OUT THE MEAN WIND SPEED, AVSPD, AND
C         DIRECTION, AVDIR
C
      DO 28 I=2,LAYTOP
      IF(IABS(IDIR(I) - IDIR(I-1)) .LT. 180)GO TO 28
      DO 27 J=1,LAYTOP
   27 IF(IDIR(J) .LT. 180)IDIR(J) = IDIR(J) + 360
      GO TO 31
   28 CONTINUE
C
   31 AVSPD = 0.0
      AVDIR = 0.0
      DO 32 I=2,LAYTOP
      IM1 = I - 1
      DALT = ALT(I) - ALT(IM1)
      AVSPD = AVSPD + 0.5 * (SPEED(I) + SPEED(IM1)) * DALT
   32 AVDIR = AVDIR + 0.5 * FLOAT(IDIR(I) + IDIR(IM1)) * DALT
C
      DO 34 I=1,LAYTOP
   34 IF(IDIR(I) .GT. 360)IDIR(I) = IDIR(I) - 360
C
      DALT = ALT(LAYTOP) - ALT(1)
      AVSPD = AVSPD/DALT
      AVDIR = AVDIR/DALT
      IF(AVDIR .GT. 180.0)GO TO 35
      AVDIR = AVDIR + 180.0
      GO TO 36
   35 AVDIR = AVDIR - 180.0
C
   36 WRITE (LUPRNT,206) AVSPD,AVDIR
C
C         IS THIS A RESEARCH, OPERATIONAL, OR PRODUCTION RUN?
```

```
      IF(IRUN .EQ. 3)GO TO 45
C
C        RESEARCH OR OPERATIONAL RUN -- ALLOW THE CALCULATED VALUE OF
C        SIGAZ TO BE CHANGED
C
      CALL DREAD(NAMEF,6,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CODE
      WRITE (ILIN23,102) SIGAZ
      CALL CHAR(0.0,YPOS,0,ILINE,43,3,0)
      CALL CHAR(352.0,YPOS,0,ILINE(23),4,0,0)
      CALL CHAR(384.0,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.0,YPOS,0,ILINE(30),6,0,0)
      IF(LVERSN) GO TO 41
      IX = 25
      READ(LU,90)IANS
      IF(IANS.EQ.1H0)STOP
      IF(IANS .NE. 1HN) GO TO 43
      IX = 30
      GO TO 43
 41   CONTINUE
      CALL IN(1,JTYPE,0.0,0.0,0,0,0,0,31,0,31,IX,IY)
 43   CONTINUE
      CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)CALL CHAR(464.0,YPOS,0,IERS,6,0,0)
      IF(IX .GT. 25)CALL CHAR(384.0,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.0
      IF(IX .LE. 25)GO TO 45
      CALL DREAD(NAMEF,7,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.0,YPOS,0,ILINE,51,3,0)
      NIN = 4
      CALL BLANK(IDATAF,2)
      CALL IN(0,JTYPE,368.0,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CODE
      READ (IDATAF,102) SIGAZ
      CALL CODE
      WRITE (ILIN24,102) SIGAZ
      CALL CHAR(0.0,YPOS,0,ILINE,50,0,0)
      YPOS = YPOS - 32.0
C
C        WRITE OUT SIGAZ, THE SIGMA OF THE WIND AZIMUTH ANGLE
C
   45 WRITE (LUPRNT,208) SIGAZ
C
      SIGAP = 0.0087266 * SIGAZ
C
C        CALCULATE THE HORIZONTAL AND VERTICAL CLOUD DIMENSIONS,
C        I.e. SIGX0 AND GSPEED
C
      SIGX0 = 0.297674 * STBALT
      GSPEED = 0.232558 * STBALT
C
C        CALCULATE AND WRITE OUT THE EFFECTIVE CLOUD HEIGHT, STBHT
C
      STBHT = STBALT
      CLDRAD = 2.15 * SIGX0
      IV2 = 0
      IF(CLDRAD+STBALT .GE. ALT(LAYTOP))IV2 = 1
      SIGZ0 = SIGX0
```

```
      IF(IV2 .EQ. 1)SIGZ0 = (ALT(LAYTOP) - STBALT + CLDRAD)/4.3
      IF(SIGZ0 .GT. 0.0)GO TO 47
      STBHT = 0.5 * ALT(LAYTOP)
      SIGZ0 = 0.64 * STBHT/2.15
      GO TO 49
   47 IF(IV2 .EQ. 1)STBHT = 0.5 * (ALT(LAYTOP) + STBALT - CLDRAD)
C
   49 WRITE (LUPRNT,209) STBHT
C
C         CALL THE CONCENTRATION PROGRAM RCONC
C
      CALL NGRAF
      CALL RWDIS(NAME,1)
      CALL EXEC(9,RCONC)
C
C         TERMINATE EXECUTION OF RCLDS
C
 1000 STOP
C
C         END OF RCLDS
C
      END
      END$
```

```
ARCLI T=A0003 IS ON CR00003 USING 00110 BLKS R=0000

FTN4X,L
      SUBROUTINE RUDIS(NAME,JJ),                          *REED3,CAB810815
C
C  ------------------------------------------------------------------
C  -                                                                -
C  -   THIS SUBROUTINE READS AND WRITES COMMON AREAS TO DISC        -
C  -   TO BE UTILIZED BY THE VARIOUS REED PROGRAMS.                 -
C  -                                                                -
C  ------------------------------------------------------------------
C
C  >>>   COMMON AREAS
C
C
C  >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C  >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISWDFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,NOHLY,NFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURFEN,FILE(3)
      INTEGER FILE
C
C  >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C  >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMONK2),ISYEAR
C
C  >>>   LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(30)
      LOGICAL LTIME
C
C  >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPT2
C
C  >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C  >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C  >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATH,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PHC,VPAR(17)),(GAMMAX,VPAR(18))
      INTEGER DCEB(144),DBUF(557)
```

```
      DIMENSION NAME(3) , ISIZE(2)
      EQUIVALENCE (OBUF(1),PI)

CBJ *** ISIZE(1) = # BLOCKS EXPECTED TO BE WRITTEN READ DURING RUN
CBJ *** ISIZE(2) = # WORDS IN THE COMMON BLOCK TO BE TRANSFERRED
      DATA ISIZE/30,557/

CBJ *** IF THE SCRATCH DATA FILE (i.e. =REED2) ISN'T THERE, CREATE IT
CBJ *** B E W A R E !!  BE SURE FILE IS PURGED BEFORE STARTING RUN !!
      CALL OPEN(ODCB,IERR,NAME,0)
      IF( (IERR.LT.0) .AND. (IERR.NE.-6) ) WRITE(6,770) IERR
770   FORMAT(" *** ATTEMPT TO OPEN, IERR =",I4)
      IF(IERR.LT.0)
     .   CALL CREAT(ODCB,IERR,NAME,ISIZE,3,0,0,144)
      IF(IERR.LE.0) WRITE(6,799) IERR
799   FORMAT(" *** CREATE/OPEN, IERR=",I4)

CBJ *** READ OR WRITE, AS PER ORDER
      IF(JJ.EQ.1)CALL WRITF(ODCB,IERR,OBUF,ISIZE(2))
      IF(JJ.EQ.0)CALL READF(ODCB,IERR,OBUF,ISIZE(2))
      CALL CLOSE(ODCB,IERR)
      RETURN
      END
      SUBROUTINE FTOP
C
C ----------------------------------------------------------------
C -                                                              -
C -    THIS SUBROUTINE COMPUTES THE INDEX OF THE TOP OF THE      -
C -    SURFACE LAYER BY FINDING SIGNIFICANT CHANGES IN THE       -
C -    SLOPE OF THE WIND DIRECTION, WIND SPEED, AND DRY          -
C -    TEMPERATURE VS. ALTITUDE CURVES                           -
C -                                                              -
C ----------------------------------------------------------------
C
C >>>   COMMON AREAS
C
C
C >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C >>>   VEHICLE PARAMETERS
      COMMON VPAR(10),SIGHCL
C
C >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNOFG,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,MONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C >>>   LAUN... TIME
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME
C
C >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SOSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONPK,DOSMAX,DOSPK,RNG3MC,RNG5MD
C
C >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATH,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
C       DIMENSION STATEMENT
C
      DIMENSION IPT(40),ITHET(40)
C
C       DATA STATEMENT FOR THE SIGNIFICANT SLOPE CHANGES FOR EACH CURVE
C
      DATA WDSLD/0.030/, WSSLD/0.015/, DTSLD/0.015/
C
C       CALCULATE THE BOTTOM OF THE CLOUD -- THE TOP OF THE
C       SURFACE LAYER WILL NEVER BE ALLOWED TO BE BELOW THE
C       BOTTOM OF THE CLOUD
C
      CLDBOT = STBALT * (1.0 - GAMMAX)
C
C       CALCULATE AND STORE THE INDEX FOR THE SIGNIFICANT SLOPE
C       CHANGES OF THE WIND DIRECTION VS. ALTITUDE CURVE
C
      DO 17 K=1,NUM
      ITHET(K) = IDIR(K)
   17 CONTINUE
      NPT = 0
      NUM1 = NUM - 1
      DELD = FLOAT(ITHET(2) - ITHET(1))
      SLOPE2 = DELD/(ALT(2) - ALT(1))
      DO 3 I=2,NUM1
      SLOPE1 = SLOPE2
      DELD = FLOAT(ITHET(I+1) - ITHET(I))
      SLOPE2 = DELD/(ALT(I+1) - ALT(I))
      IF(ALT(I) .LT. CLDBOT)GO TO 3
      SLDIF = ABS(SLOPE2 - SLOPE1)
      IF(SLDIF .LT. WDSLD)GO TO 3
      NPT = NPT + 1
      IPT(NPT) = I
    3 CONTINUE
C
C       CALCULATE AND STORE THE INDEX FOR THE SIGNIFICANT SLOPE
C       CHANGES OF THE WIND SPEED VS. ALTITUDE CURVE
C
```

```
      SLOPE2 = (SPEED(2) - SPEED(1))/(ALT(2) - ALT(1))
      DO 6 I=2,NUM1
      SLOPE1 = SLOPE2
      SLOPE2 = (SPEED(I+1) - SPEED(I))/(ALT(I+1) - ALT(I))
      IF(ALT(I) .LT. CLDBOT)GO TO 6
      SLDIF = ABS(SLOPE2 - SLOPE1)
      IF(SLDIF .LT. WSSLD)GO TO 6
      NPT = NPT + 1
      IPT(NPT) = I
    6 CONTINUE
C
C        CALCULATE AND STORE THE INDEX FOR THE SIGNIFICANT SLOPE
C        CHANGES OF THE DRY TEMPERATURE VS. ALTITUDE CURVE
C
      SLOPE2 = (TEMP(2) - TEMP(1))/(ALT(2) - ALT(1))
      DO 9 I=2,NUM1
      SLOPE1 = SLOPE2
      SLOPE2 = (TEMP(I+1) - TEMP(I))/(ALT(I+1) - ALT(I))
      IF(ALT(I) .LT. CLDBOT)GO TO 9
      SLDIF = ABS(SLOPE2 - SLOPE1)
      IF(SLDIF .LT. DTSLD)GO TO 9
      NPT = NPT + 1
      IPT(NPT) = I
    9 CONTINUE
C
C        IF THERE ARE NO SIGNIFICANT SLOPE CHANGES, MAKE THE
C        TOP OF THE LAYER TWICE THE CLOUD STABILIZATION ALTITUDE
C
      IF(NPT .GT. 0)GO TO 14
      STBAL2 = 2.0 * STBALT
      DIFMIN = 9999999.9
      DO 12 I=1,NUM
      DIF = ABS(ALT(I) - STBAL2)
      IF(DIF .GE. DIFMIN)GO TO 12
      DIFMIN = DIF
      LAYTOP = I
   12 CONTINUE
      RETURN
C
C        IF THERE IS ONLY ONE SIGNIFICANT SLOPE CHANGE, MAKE IT THE
C        TOP OF THE SURFACE LAYER
C
   14 IF(NPT .GT. 1)GO TO 16
      LAYTOP = IPT(1)
      RETURN
C
C        SORT THE SIGNIFICANT SLOPE CHANGE POINTS FROM HIGH TO LOW
C
   16 NPT1 = NPT - 1
      DO 19 I=1,NPT1
      JJ = NPT - I
      DO 18 J=1,JJ
      J1 = J + 1
      IF(IPT(J) .GE. IPT(J1))GO TO 18
      K = IPT(J)
      IPT(J) = IPT(J1)
      IPT(J1) = K
   18 CONTINUE
   19 CONTINUE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C        FIRST CHECK FOR SIGNIFICANT SLOPE CHANGES FROM ALL THREE
C        CURVES AT THE SAME POINT -- NEXT CHECK FOR SIGNIFICANT
C        SLOPE CHANGES FROM TWO OUT OF THE THREE CURVES AT THE SAME
C        POINT -- THE FIRST POINT MEETING ONE OF THESE REQUIREMENTS
C        WILL BE THE TOP OF THE SURFACE LAYER
C
      DO 27 I=1,2
      JJ = 4 - I
      J1 = 1
      DO 25 J=2,NPT
      IF(IPT(J) .EQ. IPT(J-1))GO TO 22
      J1 = 1
      GO TO 25
   22 J1 = J1 + 1
      IF(J1 .NE. JJ)GO TO 25
      LAYTOP = IPT(J)
      RETURN
   25 CONTINUE
   27 CONTINUE
C
C        THERE ARE NO DUPLICATE POINTS OF SIGNIFICANT SLOPE CHANGE,
C        MAKE THE TOP OF THE SURFACE LAYER THE FIRST POINT
C
      LAYTOP = IPT(1)
      RETURN
C
C        END OF FTOP
C
      END
      SUBROUTINE PTTOP
C
C   ----------------------------------------------------------------
C   -                                                              -
C   -  THIS SUBROUTINE ALLOWS THE USER TO CHOOSE THE TOP OF THE     -
C   -  SURFACE LAYER BY WRITING SIGNIFICANT CHANGES IN THE         -
C   -  SLOPE OF THE WIND DIRECTION, WIND SPEED, AND DRY            -
C   -  TEMPERATURE VS. ALTITUDE CURVES TO THE CRT OR PLASMASCOPE-
C   -                                                              -
C   ----------------------------------------------------------------
C
C   >>>   COMMON AREAS
C
C
C   >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C   >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C   >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNOFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,MONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C   >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C   >>>   CALCULATION TIME
```

```
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMONK2),ISYEAR
C
C >>>   LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME
C
C >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATH,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
C         DIMENSION STATEMENT
C
      DIMENSION IPT(40),DWDS(40),WSDS(40),TDS(40),ITHET(40),
     .          DSTAR(30),SSTAR(30),TSTAR(30)
C
C         DATA STATEMENT FOR THE SIGNIFICANT SLOPE CHANGES FOR EACH CURVE
C
      DATA WDSLD/0.030/, WSSLD/0.015/, DTSLD/0.015/
      DATA ISTR/1H*/
C
C         CALCULATE THE BOTTOM OF THE CLOUD -- THE TOP OF THE
C         SURFACE LAYER WILL NEVER BE ALLOWED TO BE BELOW THE
C         BOTTOM OF THE CLOUD
C
      CLDBOT = STBALT * (1.0 - GAMMAX)
      STBAL2 = 2.0 * STBALT
C
C ZERO OUT THE SIGNIFICANT SLOPE ARRAYS
C
      DO 5 K=1,NUM
      DWDS(K) = 0.0
      WSDS(K) = 0.0
      TDS(K) = 0.0
      DSTAR(K) = 0.0
      SSTAR(K) = 0.0
    5 TSTAR(K) = 0.0
C
C         CALCULATE AND STORE THE INDEX FOR THE SIGNIFICANT SLOPE
C         CHANGES OF THE WIND DIRECTION VS. ALTITUDE CURVE
C
      DO 17 K=1,NUM
      ITHET(K) = ID.....)
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
   17 CONTINUE
      NPT = 0
      NUM1 = NUM - 1
      DELD = FLOAT(ITHET(2) - ITHET(1))
      IF(DELD.GT.180.) DELD = 360. - DELD
      IF(DELD.LT.-180.) DELD = -360. - DELD
      SLOPE2 = DELD/(ALT(2) - ALT(1))
C
      DO 3 I=2,NUM1
      SLOPE1 = SLOPE2
      DELD = FLOAT(ITHET(I+1) - ITHET(I))
      IF(DELD.GT.180.) DELD = 360. - DELD
      IF(DELD.LT.-180.) DELD = -360. - DELD
      SLOPE2 = DELD/(ALT(I+1) - ALT(I))
      IF(ALT(I) .LT. CLDBOT .OR. ALT(I) .GT. STBAL2) GO TO 3
      SLDIF = (SLOPE2 - SLOPE1)
      IF(ABS(SLDIF).LT. 0.01)GO TO 3
      IF(SLOPE2.GT.0.0.AND.SLOPE1.LT.0.0) DSTAR(I) = SLDIF
      IF(SLOPE1.GT.0.0.AND.SLOPE2.LT.0.0) DSTAR(I) = SLDIF
      NPT = NPT + 1
      IPT(NPT) = I
      DWDS(I) = SLDIF
    3 CONTINUE
C
C        CALCULATE AND STORE THE INDEX FOR THE SIGNIFICANT SLOPE
C        CHANGES OF THE WIND SPEED VS. ALTITUDE CURVE
C
      SLOPE2 = (SPEED(2) - SPEED(1))/(ALT(2) - ALT(1))
      DO 6 I=2,NUM1
      SLOPE1 = SLOPE2
      SLOPE2 = (SPEED(I+1) - SPEED(I))/(ALT(I+1) - ALT(I))
      IF(ALT(I) .LT. CLDBOT .OR. ALT(I) .GT. STBAL2) GO TO 6
      SLDIF = (SLOPE2 - SLOPE1)
      IF(ABS(SLDIF).LT. 0.001)GO TO 6
      IF(SLOPE1.GT.0.0.AND.SLOPE2.LT.0.0) SSTAR(I) = SLDIF
      IF(SLOPE2.GT.0.0.AND.SLOPE1.LT.0.0) SSTAR(I) = SLDIF
      WSDS(I) = SLDIF
      NPT = NPT + 1
      IPT(NPT) = I
    6 CONTINUE
C
C        CALCULATE AND STORE THE INDEX FOR THE SIGNIFICANT SLOPE
C        CHANGES OF THE POTENTIAL TEMPERATURE VS. ALTITUDE CURVE
C
      SLOPE2 = (TEMP(2) - TEMP(1))/(ALT(2) - ALT(1))
      DO 9 I=2,NUM1
      SLOPE1 = SLOPE2
      SLOPE2 = (TEMP(I+1) - TEMP(I))/(ALT(I+1) - ALT(I))
      IF(ALT(I) .LT. CLDBOT .OR. ALT(I) .GT. STBAL2) GO TO 9
      SLDIF = (SLOPE2 - SLOPE1)
      IF(ABS(SLDIF) .LT. 0.01)GO TO 9
      IF(SLOPE1.GT.0.0.AND.SLOPE2.LT.0.0) TSTAR(I) = SLDIF
      IF(SLOPE2.GT.0.0.AND.SLOPE1.LT.0.0) TSTAR(I) = SLDIF
      NPT = NPT + 1
      IPT(NPT) = I
      TDS(I) = SLDIF
    9 CONTINUE
C
C        IF THERE ARE NO SIGNIFICANT SLOPE CHANGES, MAKE THE
C        TOP OF THE LAYER TWICE THE CLOUD STABILIZATION ALTITUDE
```

```
C
      IF(NPT .GT. 0)GO TO 14
      DIFMIN = 9999999.9
      DO 12 I=1,NUM
      DIF = ABS(ALT(I) - STBAL2)
      IF(DIF .GE. DIFMIN)GO TO 12
      DIFMIN = DIF
      LAYTOP = I
   12 CONTINUE
      WRITE (LU,100) ALT(LAYTOP)
  100 FORMAT(" NO SIGNIFICANT SLOPE CHANGES, THE TOP OF THE LAYER "
     .        "IS",F16.2)
      RETURN
C
C         IF THERE IS ONLY ONE SIGNIFICANT SLOPE CHANGE, MAKE IT THE
C         TOP OF THE SURFACE LAYER
C
   14 IF(NPT .GT. 1)GO TO 16
      LAYTOP = IPT(1)
      WRITE (LU,101) ALT(LAYTOP)
  101 FORMAT(" ONLY ONE SIGNIFICANT SLOPE CHANGE, THE TOP OF THE LAYER"
     .        "IS",F10.2)
      RETURN
C
C         SORT THE SIGNIFICANT SLOPE CHANGE POINTS FROM HIGH TO LOW
C
   16 NPT1 = NPT - 1
      DO 19 I=1,NPT1
      JJ = NPT - I
      DO 18 J=1,JJ
      J1 = J + 1
      IF(IPT(J) .GE. IPT(J1))GO TO 18
      K = IPT(J)
      IPT(J) = IPT(J1)
      IPT(J1) = K
   18 CONTINUE
   19 CONTINUE
C
C WRITE TO THE USER THE ALTITUDES HAVING SIGNIFICANT SLOPE CHANGES
C
      WRITE (LU,102)
  102 FORMAT(/" LEVEL  ALT       WIND DIR      WIND SPEED    TEMPERATURE"/
     .        "                  {)A v SLOPE      {)A v SLOPE"    .
     .        "      {)A V SLOPE")
C
C** WRITE THE HEADER TO THE PRINTER.
C
      WRITE (6,113)
  113 FORMAT(/" LEVEL  ALT       WIND DIR      WIND SPEED    TEMPERATURE"/
     .        "                 SLOPE DIFF    SLOPE DIFF"
     .        "     SLOPE DIFF")
      DO 33 K=1,NUM
      IF(ALT(K).LT.CLDBOT.OR.ALT(K).GT.STBAL2) GO TO 33
      IF(DWDS(K).GT.1000.) DWDS(K) = 99.9999
      IF(WSDS(K).GT.1000.) WSDS(K) = 99.9999
      IF(TDS(K).GT.1000.) TDS(K) = 99.9999
      IF(DWDS(K).NE.0.0) GO TO 37
      IF(WSDS(K).NE.0.0) GO TO 37
      IF(TDS(K).EQ.0.0) GO TO 33
[illegible]
```

```
     .WRITE (LU,103) K,ALT(K),DWDS(K),WSDS(K),TDS(K)
      IF(DSTAR(K).NE.0.0.AND.SSTAR(K).NE.0.0.AND.TSTAR(K).EQ.0.0)
     .WRITE (LU,104) K,ALT(K),DWDS(K),ISTR,WSDS(K),ISTR,TDS(K)
      IF(DSTAR(K).NE.0.0.AND.SSTAR(K).NE.0.0.AND.TSTAR(K).NE.0.0)
     .WRITE (LU,105) K,ALT(K),DWDS(K),ISTR,WSDS(K),ISTR,TDS(K),ISTR
      IF(DSTAR(K).NE.0.0.AND.SSTAR(K).EQ.0.0.AND.TSTAR(K).EQ.0.0)
     .WRITE (LU,106) K,ALT(K),DWDS(K),ISTR,WSDS(K),TDS(K)
      IF(DSTAR(K).EQ.0.0.AND.SSTAR(K).NE.0.0.AND.TSTAR(K).EQ.0.0)
     .WRITE (LU,107) K,ALT(K),DWDS(K),WSDS(K),ISTR,TDS(K)
      IF(DSTAR(K).EQ.0.0.AND.SSTAR(K).EQ.0.0.AND.TSTAR(K).NE.0.0)
     .WRITE (LU,108) K,ALT(K),DWDS(K),WSDS(K),TDS(K),ISTR
      IF(DSTAR(K).EQ.0.0.AND.SSTAR(K).NE.0.0.AND.TSTAR(K).NE.0.0)
     .WRITE (LU,109) K,ALT(K),DWDS(K),WSDS(K),ISTR,TDS(K),ISTR
      IF(DSTAR(K).NE.0.0.AND.SSTAR(K).EQ.0.0.AND.TSTAR(K).NE.0.0)
     .WRITE (LU,110) K,ALT(K),DWDS(K),ISTR,WSDS(K),TDS(K),ISTR
C
C** WRITE THE CRT DISPLAY TO THE PRINTER.
C
      IF(DSTAR(K).EQ.0.0.AND.SSTAR(K).EQ.0.0.AND.TSTAR(K).EQ.0.0)
     .WRITE (6,103) K,ALT(K),DWDS(K),WSDS(K),TDS(K)
      IF(DSTAR(K).NE.0.0.AND.SSTAR(K).NE.0.0.AND.TSTAR(K).EQ.0.0)
     .WRITE (6,104) K,ALT(K),DWDS(K),ISTR,WSDS(K),ISTR,TDS(K)
      IF(DSTAR(K).NE.0.0.AND.SSTAR(K).NE.0.0.AND.TSTAR(K).NE.0.0)
     .WRITE (6,105) K,ALT(K),DWDS(K),ISTR,WSDS(K),ISTR,TDS(K),ISTR
      IF(DSTAR(K).NE.0.0.AND.SSTAR(K).EQ.0.0.AND.TSTAR(K).EQ.0.0)
     .WRITE (6,106) K,ALT(K),DWDS(K),ISTR,WSDS(K),TDS(K)
      IF(DSTAR(K).EQ.0.0.AND.SSTAR(K).NE.0.0.AND.TSTAR(K).EQ.0.0)
     .WRITE (6,107) K,ALT(K),DWDS(K),WSDS(K),ISTR,TDS(K)
      IF(DSTAR(K).EQ.0.0.AND.SSTAR(K).EQ.0.0.AND.TSTAR(K).NE.0.0)
     .WRITE (6,108) K,ALT(K),DWDS(K),WSDS(K),TDS(K),ISTR
      IF(DSTAR(K).EQ.0.0.AND.SSTAR(K).NE.0.0.AND.TSTAR(K).NE.0.0)
     .WRITE (6,109) K,ALT(K),DWDS(K),WSDS(K),ISTR,TDS(K),.STR
      IF(DSTAR(K).NE.0.0.AND.SSTAR(K).EQ.0.0.AND.TSTAR(K).NE.0.0)
     .WRITE (6,110) K,ALT(K),DWDS(K),ISTR,WSDS(K),TDS(K),ISTR
  103 FORMAT(I6,F8.1,F12.5,2(F13.5))
  104 FORMAT(I6,F8.1,F12.5,A1,F11.5,A1,F12.5)
  105 FORMAT(I6,F8.1,F12.5,A1,F11.5,A1,F11.5,A1)
  106 FORMAT(I6,F8.1,F12.5,A1,F12.5,F13.5)
  107 FORMAT(I6,F8.1,F12.5,F13.5,A1,F12.5)
  108 FORMAT(I6,F8.1,F12.5,F13.5,F13.5,A1)
  109 FORMAT(I6,F8.1,F12.5,F13.5,A1,F12.5,A1)
  110 FORMAT(I6,F8.1,F12.5,A1,F12.5,F13.5,A1)
   33 CONTINUE
      WRITE (LU,112)
      WRITE (6,112)
  112 FORMAT(" * INDICATES A REVERSAL IN THE SIGN OF THE SLOPE")
   34 WRITE (LU,111)
  111 FORMAT(" {&dJ ENTER THE NUMBER OF THE LAYER FOR THE TOP {&d@"
     .        "{&dJ OF THE SURFACE LAYER{&d@")
      READ (LU,*) LAYER
      LAYTOP = LAYER
      RETURN
C
C        END OF PTTOP
C
      END
      SUBROUTINE DREAD(NAMEF,LNUM,ILINE)
C     ------------------------------------------------------------------
C     -
```

```
C  -  THIS SUBROUTINE READS A SPECIFIED QUESTION FROM A SPECIFIED  -
C  -  QUESTION FILE. THE QUESTION IS THEN DISPLAYED UPON THE        -
C  -  PLASMASCOPE FOR PROCESSING.                                   -
C  -                                                               -
C  -----------------------------------------------------------------
C
      DIMENSION NAMEF(3),IDCB(144),IBUF(40),ILINE(32)
      CALL OPEN(IDCB,IERR,NAMEF,0)
      IF(IERR .LE. 0) WRITE(6,799) IERR, NAMEF
 799  FORMAT(" ERROR ",I5, " ON FILE ",3A2)
      LOOP = LNUM - 1
      DO 10 I=1,LOOP
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
   10 CONTINUE
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ (IBUF,100) (ILINE(I),I=1,32)
  100 FORMAT(32A2)
      CALL CLOSE(IDCB,IERR)
      RETURN
      END
      SUBROUTINE BLANK(IBUF,II)
C
C  -----------------------------------------------------------------
C  -                                                               -
C  -  THIS SUBROUTINE FILLS A SPECIFIED ARRAY WITH A SPECIFIED     -
C  -  NUMBER OF BLANKS.                                            -
C  -                                                               -
C  -----------------------------------------------------------------
C
      DIMENSION IBUF(40)
      DATA IBLK/2H  /
      DO 10 I=1,II
 10   IBUF(I) = IBLK
      RETURN
      END
      SUBROUTINE LERS(YPOS)
C
C  -----------------------------------------------------------------
C  -                                                               -
C  -  THIS SUBROUTINE ERASES A SPECIFIED LINE OF 64-CHARACTERS     -
C  -  FROM THE PLASMASCOPE.                                        -
C  -                                                               -
C  -----------------------------------------------------------------
C
      DIMENSION IERS(32)
      DATA IERS/32*2H  /
      IF(YPOS .GT. 48.0)GO TO 4
      YPOS = 458.0
      CALL CLEAR
      RETURN
    4 CALL CHAR(0.0,YPOS,0,IERS,64,0,0)
      CALL CHAR(0.0,YPOS-16.0,0,IERS,64,0,0)
      RETURN
      END
      SUBROUTINE RSGAZ(J1,J2,J3,RSIG)

   -----------------------------------------------------------------
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
C -                                                   -
C -  THIS SUBROUTINE CALCULATES A SIGMA VALUE GIVEN   -
C -  ALTITUDE, SPEED, TEMP, AND PRESSURE FOR THE      -
C -  FIRST LEVEL OF DATA, THE 1000FT LEVEL OF DATA    -
C -  AND THE 1000MB LEVEL OF DATA
C -                                                   -
C ------------------------------------------------------
C
C >>>   COMMON AREAS
C
C
C >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C >>>   OPTION PARAMETERS
      COMMON [illegible]OS,YES,ISNDFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       [illegible]N,LU,NUM,MONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER
C
C >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMOI(2),ISYEAR
C
C >>>   LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME
C
C >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATH,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PHO,VPAR(17)),(GAMMAX,VPAR(18))
      DATA C1,C2,C3,C4,C5,C6/-.008,-.00175,.0008,.50864522,.1132,
     . 3.8163/
      [illegible]TA C[illegible].0297
C CALCULATION OF SIGAR
C NEWTONS METHOD FOR SOLUTION OF F([illegible],[illegible],[illegible]) = 0
```

```
      F(X,B,D) =(1.-X**4)/(16.*X**2*(ALOG(D)+C4-2.*ALOG(1.+X)
     . - ALOG(1.+X**2)+2.*ATAN(X))**2) - B
      FP(X,D) =(-X**4-1.)/(8.*X**3*(ALOG(D)+C4-2.*ALOG(1.+X)
     . - ALOG(1.+X**2)+2.*ATAN(X))**2) + (1.-X**4)/(2.*(1.+X)
     . *(1.+X**2)*(ALOG(D)+C4-2.*ALOG(1.+X)-ALOG(1.+X**2)+
     . 2.*ATAN(X))**3)
C
C*** READ 1ST DATA LEVEL
C
      Z1 = ALT(J1)
      V1 = SPEED(J1)
      T1 = TEMP(J1)
      PZ1 = PRESS(J1)
C
C*** READ SECOND DATA LEVEL
C
      Z2 = ALT(J2)
      V2 = SPEED(J2)
      T2 = TEMP(J2)
      PZ2 = PRESS(J2)
C
C*** IF TOWER DATA IS NOT BEING USED, INTERPOLATE TO FIND THE 1000
C*** FOOT LEVEL DATA
C
      Z3 = ALT(J3)
      T3 = TEMP(J3)
      PZ3 = PRESS(J3)
      WRITE (6,613) PZ1,PZ2,PZ3
  613 FORMAT(" PRESSURES ARE:",3F10.2)
      IF(ITDU.EQ.1) GO TO 5
C
      FAC = (304.8 - ALT(J3))/(ALT(J3+1)-ALT(J3))
      Z3 = ALT(J3)+(FAC*(ALT(J3+1)-ALT(J3)))
      T3 = TEMP(J3) + FAC*(TEMP(J3+1)-TEMP(J3))
      PZ3 = PRESS(J3) + FAC*(PRESS(J3+1)-PRESS(J3))
C
    5 T1 = T1+273.16
      T2 = T2+273.16
      T3 = T3+273.16
C
C*** INITIALIZE Z0
C
      Z0 = .20
C PZ1 AND PZ3 IN MILLIBARS
C V1,V2 AND V3 IN METER/SEC
C Z1,Z2 AND Z3 IN METERS
C T1,T2 AND T3 IN DEG K
C Z0 IN METERS
      E = 22.9183118
      V=V2
      T=(T1+T2+T3)/3.
C
C*** E IS A CONSTANT INCORPORATING VON KARMAN'S CONSTANT AND
C*** ALSO CONVERTS FROM RADIANS TO DEGREES.
C
C*** CALCULATE GEOMETRIC MEAN HEIGHT, AVERAGE THETA, AVERAGE HEIGHT
C
      Z=(Z1+Z2+Z3)**.33333
      THETA1 [illegible] T1*(([illegible]./P[illegible] *.28[illegible]
      THETA2 = T2
```

```
      THETA3 = T3*((1000./PZ3)**.288)
      ZA = (Z1+Z2+Z3)/3.
      THETAA = (THETA1 + THETA2 + THETA3)/3.
      D = Z/Z0
      Z0Z0 = ALOG(D)
C
C*** CALCULATE VERTICAL GRADIENT OF POTENTIAL TEMPERATURE
C
      DZTHET = ((Z1-ZA)*(THETA1-THETAA)+(Z2-ZA)*(THETA2-THETAA)
     .         +(Z3-ZA)*(THETA3-THETAA))/((Z1-ZA)**2 + (Z2-ZA)**2
     .         +(Z3-ZA)**2)
C
C*** CALCULATE B, THE STABILITY RATIO
C*** B.LT.0 = UNSTABLE
C*** B.GT.0 = STABLE
C
      B = 9.8*DZTHET*Z**2/(T*V**2)
      IF(B.LT.0) GO TO 2
      IF(B.GT.0) GO TO 6
      RSIG = 40.016/ALOG(D)
      GO TO 1000
C
C*** USE NEWTON'S METHOD TO SOLVE EQUATION 14 (REPORT 2923)
C
    2 R = 1.5
      U = F(R,B,D)
      DO 3 I = 1,50
      R1 = R - F(R,B,D)/FP(R,D)
      U=F(R1,B,D)
      IF(ABS(R1-R).LT.1.E-7) GO TO 21
      IF(I.EQ.49) USAV = U
      IF(I.NE.50) GO TO 3
      IF(USAV.LT.0..AND.U.GT.0..OR.USAV.GT.0..AND.U.LT.0.) GO TO 21
    3 R = R1
      RSIG = 30.
      GO TO 1000
   21 R11 = (1.-R1**4)/16.
      A = Z0Z0 +C4-2.*ALOG(1.+R1)-ALOG(1.+R1**2)+2.*ATAN(R1)
C
C*** ASSIGN THE CORRECT F(B) FOR EACH VALUE OF B ACCORDING TO
C*** TABLE 6 IN REPORT 2923
C
      IF(B.LT.C1) RSIG = E*2.7/A
      IF(B.GE.C1.AND.B.LT.C2) GO TO 23
      IF(B.GE.C2) GO TO 24
      GO TO 1000
C
C*** COMPUTE SIGR ACCORDING TO EQUATION 1 IN REPORT 2923 FOR
C*** UNSTABLE CONDITIONS
C
23    FB2 = 2.7 + 112.*(-C1 + B)
      RSIG = E*FB2/A
      GO TO 1000
24    FB3 = 3.4 - 725.5*(-C2 +B)
      RSIG = E*FB3/A
      GO TO 1000
C
C*** FOR STABLE CONDITIONS, USE EQUATION 3(A) IN REPORT 2923 TO
C*** CALCULATE RICHARDSON NUMBER
C
```

```
 6      AP = ZO20 + 1.
        A1 = 7.*SQRT(B)*AP
        A2 = 1.
        A3 = -SQRT(B)*(AP-1.)
        RAD = A2**2 - 4.*A1*A3
 C
 C*** CHECK SIGN OF THE RADICAL, USE VALUE OF RSGIG = 30. FOR
 C***                              IMAGINARY VALUES.
 C
        RSIG = 30.0
        IF(RAD.LT.0.0) GO TO 1000
        IF(RAD.GT.0.0) GO TO 90
        RE11 = -A2/(2.*A1)
        S1 = 1. - 7.*RE11**2
        RI3 = (S1-1.)/(-7.)
 C
 C*** ASSIGN APPROPRIATE VALUE OF FB ACCORDING TO TABLE 6 IN REPORT 2923
 C
        IF(B.LT.C3) GO TO 27
        GO TO 28
 C
 C*** FIND RSIG FROM EQUATION 1 IN REPORT 2923 FOR STABLE CONDITIONS
 C
 27     FB3 = 3.4 - 725.5*(-C2 + B)
        RSIG = (E*FB3)/(ZO20 - 7.*RI3/(1. - 7.*RI3))
        SIGR20=(E*FB3)/(C6+ZO20)
        IF(RI3.GE.C5) RSIG = SIGR20
        GO TO 1000
 28     FB4 = 1.55 + 38.04*(B - .0008)
        FB5 = 2.35 + 5.43*(B - C7)
        RSIG = (E*FB4)/(ZO20 - 7.*RI3/(1. - 7.*RI3))
        IF(B.GE.C7)RSIG = (E*FB5)/(ZO20 - 7.*RI3/(1. - 7.*RI3))
        SIGR21 = (E*FB4)/(C6 + ZO20)
        SIGR22 = (E*FB5)/(C6 + ZO20)
        IF(RI3.GE.C5.AND.B.LT.C7)RSIG = SIGR21
        IF(RI3.GE.C5.AND.B.GE.C7)RSIG = SIGR22
        GO TO 1000
 90     RE1 = (-A2 + SQRT(RAD))/(2.*A1)
        RI4 = RE1**2
 C
 C*** ASSIGN PROPER VALUE OF FB ACCORDING TO TABLE 6 IN REPORT 2923
 C
        IF(B.LT.C3) GO TO 37
        GO TO 38
 C
 C*** FIND THE RICHARDSON NUMBER USING EQUATION 2(A) FOR UNSTABLE
 C*** CONDITIONS
 C
 37     FB3 = 3.4 - 725.5*(-C2+B)
        RSIG = (E*FB3)/(ZO20 - 7.*RI4/(1. - 7.*RI4))
        SIGR20=(E*FB3)/(C6+ZO20)
        IF(RI4.GE.C5) RSIG = SIGR20
        GO TO 1000
 38     FB4 = 1.55 + 38.04*(B - .0008)
        FB5 = 2.35 + 5.43*(B - C7)
        RSIG = (E*FB4)/(ZO20 - 7.*RI4/(1. - 7.*RI4))
        IF(B.GE.C7)RSIG = (E*FB5)/(ZO20 - 7.*RI4/(1. -7.*RI4))
        SIGR21=(E*FB4)/(C6+ZO20)
        SIGR22 = (E*FB5)/(C6+ZO20)
        IF(RI4.GE.C5.AND.B.LT.C7) RSIG = SIGR21
```

```
      IF(R14.GE.C5.AND.B.GE.C7) RSIG = SIGR22
C
C*** CHECK FOR VALID SIGA VALUE
C
1000  IF (RSIG.LE.0.  OR. RSIG.GT.30.) RSIG = 30.
      RETURN
      END
      END$
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
&RMTFS T-00004 IS ON CR00003 USING 00694 BLKS R-0000

FTN4
      PROGRAM RMTPS(3),REVISED MET PLOT PROG                  *CAB,810813
C     ******************************************************************
C     *                                                                *
C     *  THIS PROGRAM GENERATES A METEORLOGICAL PROFILE OF A SOUNDING  *
C     *  UPON THE PLASMASCOPE WITH THE EFFECTIVE CLOUD HEIGHT DRAWN    *
C     *  AS AN ELLIPSE JUST BELOW THE TOP OF THE SURFACE LAYER.        *
C     *                                                                *
C     ******************************************************************
C  >>>   COMMON AREAS
C  >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C  >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNDFD,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,MONLY,MFORM,LUPRNT.ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C  >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C  >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C  >>>   LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME, LVRCRT, LVRPLS
C
C  >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPT2
C
C  >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SGSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C  >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C  >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)) (PHO,VPAR(17)),(GAMMAX,VPAR(18))
C
C     FORMAT STATEMENTS
 2000 FORMAT (I2,1XA2.A1,1XI4)
   01 ...MAT ...)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
 2002 FORMAT (F6.1)
 2003 FORMAT (F4.1)
C
C          TYPE AND DIMENSION STATEMENTS
         INTEGER RMETQ(3)
         DIMENSION WSX(30),DTX(30),PTX(30),WDX(30),CURVEY(30)
         DIMENSION XAX(3),YAX(3),XCD(3),YCD(3),IFOUR(2)
         DIMENSION IALTL(8),TSURX(20),BSURX(10)
         DIMENSION IALTCH(336),IHARD(16)
         DIMENSION IXNUM(13),IYNUM(22),IALTCI(8),IDIM(10)
         DIMENSION IDATL(2),ITIML(2),ICWD(9)
         DIMENSION AWDIR(30),APTEMP(30)
         DIMENSION XL(2),YL(2),IDT(12),IPT(11),IWS(8),IWD(10)
         DIMENSION ISURL1(30),IALTSP(8)
         DIMENSION ISURT(22),ISURT1(16),IALTP(8),IALTCL(8)
         DIMENSION ICRVT(4),ISTL(12),XWD(18)
         DIMENSION IWDL(18,4),IWDL1(18),IWDL2(18),IWDL3(18),IWDL4(18)
         DIMENSION IMET(2),INSTAL(2,2),ISTAB(4),ITOP(2),IBOT(2),IALPHA(6),
        .          NAME(3),ISIGMA(3),XELL(100),YELL(100),NAFIL(5)
C
C                   EQUIVALENCE STATEMENT
C
         EQUIVALENCE (IWDL1(1),IWDL(1,1)),(IWDL2(1),IWDL(1,2)),
        .            (IWDL3(1),IWDL(1,3)),(IWDL4(1),IWDL(1,4))
C
C          DATA STATEMENTS
C
         DATA IHARD/2HHA,2HRD,2H C,2HOP,2HY ,2HDE,2HSI,2HRE,2HD?,2H  ,
        .2H  ,2HYE,2HS ,2H  ,2H  ,2HH0/
         DATA RMETQ/2HRM,2HTQ,1H2/
         DATA NAME/2H-R,2HEE,2HD2/
         DATA IWDL1/2H 0,2H  ,2H  ,2H  ,2H 9,2H0 ,2H  ,2H  ,2H18,2H0 ,
        .           2H  ,2H  ,2H27,2H0 ,2H  ,2H  ,2H36,2H0 /
         DATA IWDL2/2H90,2H  ,2H  ,2H  ,2H18,2H0 ,2H  ,2H  ,2H27,2H0 ,
        .           2H  ,2H  ,2H36,2H0 ,2H  ,2H  ,2H90,2H  /
         DATA IWDL3/2H18,2H0 ,2H  ,2H  ,2H27,2H0 ,2H  ,2H  ,2H36,2H0 ,
        .           2H  ,2H  ,2H90,2H  ,2H  ,2H  ,2H18,2H0 /
         DATA IWDL4/2H27,2H0 ,2H  ,2H  ,2H36,2H0 ,2H  ,2H  ,2H90,2H  ,
        .           2H  ,2H  ,2H18,2H0 ,2H  ,2H  ,2H27,2H0 /
         DATA XWD/300.,300.,320.,320.,340.,340.,360.,360.,380.,380.,
        .         400.,400.,420.,420.,440.,440.,460.,460./
         DATA ICRVT/2HWS,2HDT,2HPT,2HWD/
         DATA IEXP3/2H3 /
         DATA ISTL/2HSP,2HEE,2HD(,2HM/,2HS),2H  ,2HTE,2HMP,2H(D,2HEG,2H C,
        12H) /
         DATA ISURL1/2HSU,2HRF,2HAC,2HE),2H  ,
        .            2H  ,2HPR,2HES,2HSU,2HRE,
        .            2H  ,2H  ,2H  ,2H  ,2H M,
        .            2HB ,2H  ,2H  ,2H D,2HEN,
        .            2HSI,2HTY,2H  ,2H  ,2H  ,
        .            2H  ,2H C,2H/M,2H  ,2H  /
         DATA IDT/2HDR,2HY ,2HTE,2HMP,2HER,2HAT,2HUR,2HE ,2H(D,2HEG,
        12H C,2H) /
         DATA IPT/2HPO,2HTE,2HNT,2HIA,2HL ,2HTE,2HMP,2H (,2HDE,2HG ,
        12HC)/
         DATA IMINUS/1H-/
         DATA IWS/2HWI,2HND,2H S,2HPE,2HED,2H (,2HM/,2HS)/
         DATA IWD/2HWI,2HND,2H D,2HIR,2HEC,2HTI,2HON,2H (,2HDE,2HG)/
         DATA ISURT/2HSU,2HRF,2HAC,2HE ,2H  ,2H  ,2H  ,2H  ,2HTA,2HP ,
        .           2HLA,2HYE,2HP ,2H  ,2H  ,2H  ,2H  ,2HBO,2HT ,2HLO,
```

```
      .          2HYE,2HR /
       DATA ISURTI/2HSU,2HRF,2HAC,2HE ,2H  ,2HTO,2HP ,2HLA,2HYE,2HR ,
      .           2H  ,2HBO,2HT ,2HLA,2HYE,2HR /
       DATA IDATL/2HDA,2HTE/,ITIML/2HTI,2HME/
       DATA IALTL/2H A,2H L,2H T,2H I,2H T,2H U,2H D,2H E/
       DATA TSURX/100.,130.,140.,170.,180.,210.,220.,250.,260.,290.,
      1300.,330.,340.,370.,380.,410.,420.,450.,460.,490./
       DATA BSURX/100.0,160 0,182.5,242.5,265.0,325.0,347.5,407.5,430.0,
      .          490.0/
       DATA ITOP/2H T,2HOP/, IBOT/2H B,2HOT/
       DATA IXNUM/2H10,2H-5,2H 0,2H 5,2H10,2H15,2H20,2H25,2H30,2H35,
      .          2H40,2H45,2H50/
       DATA XAX/460.0,100.0,100.0/, YAX/90.0,90.0,410.0/
       DATA YCD/90.,90.,410./
       DATA IYNUM/2H  ,2H 0,2H 4,2H00,2H 8,2H00,2H12,2H00,2H16,2H00,
      12H20,2H00,2H24,2H00,2H28,2H00,2H32,2H00,2H36,2H00,2H40,2H00/
       DATA IDIM/2H16,2H00,2H12,2H00,2H 8,2H00,2H 4,2H00,2H  ,2H0 ,
      .          2H 4,2H00,2H 8,2H00,2H12,2H00,2H16,2H00/
       DATA ICUD/2HCR,2HOS,2HS ,2HWI,2HND,2H D,2HIM,2H (,2HM)/
       DATA IMET/2H(M,1H)/, INSTAL/2HVA,2HFB,2HKS,1HC/
       DATA ISTAB/2HST,2HAB,2H H,2HT:/, ISIGMA/2HSI,2HGM,2HA:/
       DATA IFOUR/2H40,2H00/
       DATA NAFIL/2HFI,2HLE,2H U,2HSE,2HD:/
C
C         THE FOLLOWING DATA STATEMENT CONTAINS OCTAL REPRESENTATION
C         OF AN ALTERNATE CHARACTER SET (5 BY 6 RASTER UNITS IN SIZE)
C         AS FOLLOWS: 0-9,A-Z, AND SPECIAL CHARACTERS +,-,*,/,(,)
C
       DATA IALTCH/36B,41B,41B,36B,4*0,0,21B,77B,1B,4*0,                0,1
      1            23B,45B,45B,31B,4*0,42B,41B,51B,66B,4*0,             2,3
      1            14B,24B,77B,4B,4*0,72B,51B,51B,46B,4*0,              4,5
      1            36B,45B,45B,2B,4*0,60B,43B,44B,70B,4*0,              6,7
      1            26B,51B,51B,26B,4*0,20B,51B,51B,36B,4*0,             8,9
      1            37B,50B,50B,37B,4*0,77B,51B,51B,26B,4*0,             A,B
      1            36B,41B,41B,    22B,4*0,77B,41B,41B,36B,4*0,         C,D
      1            77B,51B,51B,41B,4*0,77B,50B,50B,40B,4*0,             E,F
      1            36B,41B,45B,26B,4*0,77B,10B,10B,77B,4*0,             G,H
      1            0,41B,77B,41B,4*0,42B,41B,76B,40B,4*0,               I,J
      1            77B,14B,    22B,41B,4*0,77B,1B,1B,1B,4*0,            K,L
      1            77B,20B,10B,20B,77B,3*0,77B,30B,6B,77B,4*0,          M,N
      1            36B,41B,41B,36B,4*0,77B,44B,44B,30B,4*0,             O,P
      1            34B,42B,42B,35B,4*0,77B,44B,46B,31B,4*0,             Q,R
      1            22B,51B,45B,22B,4*0,40B,40B,77B,40B,40B,3*0,         S,T
      1            76B,1B,1B,76B,4*0,74B,2B,1B,2B,74B,3*0,              U,V
      1            76B,1B,36B,1B,76B,3*0,61B,12B,04B,12B,61B,3*0,       W,X
      1            60B,10B,17B,10B,60B,3*0,41B,43B,45B,51B,61B,3*0,     Y,Z
      1            2*4B,37B,2*4B,3*0,5*4B,3*0,21B,12B,37B,12B,21B,3*0,  +,-
      1            1B,2B,4B,10B,20B,3*0,0,36B,41B,5*0,0,41B,36B,5*0/    /,(
       DATA IALTC1/0,12B,12B,12B,4*0/                                    =,SP
       DATA IALTP/0,1B,6*0/
       DATA IALTCL/0,12B,6*0/
       DATA IALTSP/8*0/
C
C         READ THE COMMON DISK FILE
C
       CALL RWDIS(NAME,0)
C
C         CALL SUBROUTINE GRAF TO INITIALIZE GRAPHIC MODE ON PLASMASCOPE
C
       CALL GRAF(1)
```

```
C
C        CALL SUBROUTINE VERSN TO DETERMINE IF RUNNING ON PLASMASCOPE
C        (IVERSN=0) OR CRT(IVERSN=1)
C
      CALL VERSN(IVERSN)
      LVRCRT = IVERSN .NE. 0
      LVRPLS = .NOT.LVRCRT
C
C        CALL SUBROUTINE CLEAR TO CLEAR SCREEN IF RUNNING ON THE PLASMASCOPE
C
      IF(LVRPLS) CALL CLEAR
C
C        LOAD THE ALTERNATE CHARACTER SET INTO THE PLASAMASCOPE
C
      CALL LALT(1H0,IALTCH,10)
      CALL LALT(1HA,IALTCH(81),26)
      CALL LALT(1H+,IALTCH(289),6)
      CALL LALT(1H=,IALTC1(1),1)
      CALL LALT(1H ,IALTSP,1)
      CALL LALT(1H:,IALTCL,1)
      CALL LALT(1H.,IALTP,1)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C        THIS SECTION DRAWS THE NET PLOT FORM
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      XSCALE = 0.6
C
C*** IF CRT VERSION IN AND FORM DRAWING NOT DESIRED, DELETE FORM DRAWING
C
      IF( LVRCRT .AND. (MFORM.EQ.0) ) GO TO 34
C
C        DRAW THE DATE LABEL
C
      CALL CHAR(20.0,490.0,0,IDATL,4,2,1)
      XL(1) = 20.0
      XL(2) = 48.0
      YL(1) = 488.0
      YL(2) = 488.0
      CALL LINE(XL,YL,2,0)
C
C        DRAW THE TIME LABEL
C
      CALL CHAR(164.0,490.0,0,ITIML,4,2,1)
      XL(1) = 164.0
      XL(2) = 192.0
      CALL LINE(XL,YL,2,0)
C
C        DRAW THE SURFACE PRESSURE, DENSITY, AND, IF REQUIRED,
C        STABILIZATION HEIGHT AND SIGMA LABELS
C
      CALL CHAR(20.0,475.0,0,ISURL1,60,2,1)
      IF(LVRPLS)CALL CHAR(468.0,478.0,0,IEXP3,1,2,1)
      IF(LVRCRT)CALL CHAR(318.0,478.0,0,IEXP3,1,2,1)
      XL(1) = 20.0
      XL(2) = 76.0
      YL(1) = 473.0
      YL(2) = 473.0
```

```
      CALL LINE(XL,YL,2,0)
      IF(LVRFLS)GO TO 3
      CALL CHAR(374.0,475.0,0,ISTAB,8,2,1)
      CALL CHAR(466.0,475.0,0,ISTL(4),1,2,1)
      XL(1) = 374.0
      XL(2) = 422.0
      CALL LINE(XL,YL,2,0)
      CALL CHAR(374.0,490.0,0,ISIGMA,6,2,1)
      XL(2) = 410.0
      YL(1) = 488.0
      YL(2) = 488.0
      CALL LINE(XL,YL,2,0)
C
C        PRINT SURFACE HEADER -- TOP LAYER HEADER -- BOT LAYER
C        HEADER (IF REQUIRED)
C
    3 IF(LVRCRT)GO TO 4
      I = 20
      IF(ICALC .EQ. 1)I = 32
      CALL CHAR(222.0,461.0,0,ISURT1,I,2,1)
      GO TO 5
    4 I = 26
      IF(ICALC .EQ. 1)I = 44
      CALL CHAR(222.0,461.0,0,ISURT,I,2,1)
    5 XL(1) = 222.0
      XL(2) = 278.0
      YL(1) = 459.0
      YL(2) = 459.0
      CALL LINE(XL,YL,2,0)
      XL(1) = 302.0
      XL(2) = 374.0
      CALL LINE(XL,YL,2,0)
      IF(ICALC .NE. 1)GO TO 8
      XL(1) = 398.0
      XL(2) = 470.0
      CALL LINE(XL,YL,2,0)
C
C        DRAW DRY TEMPERATURE LABEL
C
    8 CALL CHAR(30.0,450.0,0,IDT,24,2,1)
C
C        DRAW POTENTIAL TEMPERATURE LABEL
C
      CALL CHAR(30.0,440.0,0,IPT,22,2,1)
C
C        DRAW WIND SPEED LABEL
C
      CALL CHAR(30.0,430.0,0,IWS,16,2,1)
C
C        DRAW WIND DIRECTION LABEL
C
      CALL CHAR(30.0,420.0,0,IWD,20,2,1)
C
C        DRAW X AND Y AXIS
C
C XAX AND YAX ARE SET IN A DATA STATEMENT--THE USE OF THE XSCALE
C REDUCES THE MET PLOT ON THE X AXIS.
C
         1)     X(1)    SCA
      CALL LINE(XAX,YAX,3,0)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C          DRAW X AXIS LABELS
C
      CALL CHAR(100.0,70.0,0,ISTL,24,2,1)
C
C          DRAW TICK MARKS ON X AXIS
C XINC IS THE SPACE BETWEEN EACH TIC MARK
C
      XINC = (XAX(1) - XAX(2))/12.0
      TIC =  XAX(2) - XINC
      YL(1) = 88.0
      YL(2) = 92.0
      COORD = TIC - 8.0
      DO 10 I=1,13
      TIC = TIC + XINC
      XL(1) = TIC
      XL(2) = TIC
      CALL LINE(XL,YL,2,0)
      XL(1) = XL(1) + XINC/2.0
      XL(2) = XL(2) + XINC/2.0
      IF(I .NE. 13)CALL LINE(XL,YL,2,0)
      COORD = COORD + XINC
      IF(I .EQ. 1)CALL CHAR(84.0,80.0,0,IMINUS,1,2,1)
   10 CALL CHAR(COORD,80.0,0,IXNUM(I),2,2,1)
C
C          DRAW WIND DIRECTION AXIS
C
C XSCALE IS USED HERE TO SHIFT THE WIND DIRECTION AXIS TO THE LEFT
C
      XL(1) = 300.0 * XSCALE
      XL(2) = 460.0 * XSCALE
      YL(1) = 60.0
      YL(2) = 60.0
      CALL LINE(XL,YL,2,0)
C
C          DRAW WIND DIRECTION AXIS LABEL
C
C XSCALE IS USED TO SHIFT THE WIND DIRECTION AXIS LABLE TO THE LEFT
C
      XC = 310.0 * XSCALE
      CALL CHAR(XC,40.0,0,IWD,20,2,1)
C
C          DRAW TICK MARKS ON WIND DIRECTION AXIS
C
C DO 15 SCALES THE TIC MARKS TO CONFORM TO THE SCALED DOWN
C WIND DIRECTION AXIS AND WIND DIRECTION AXIS LABLE.
C
      YL(1) = 58.0
      YL(2) = 62.0
      DO 15 K=1,18
   15 XWD(K) = XWD(K) * XSCALE
      DO 16 I=1,9
      N = 2 * I - 1
   16 CALL LINE(XWD(N),YL,2,0)
C
C          DRAW TICK MARKS ON Y AXIS
C
      TIC = 58.0
      XL(1) = 98.0
      XL(2) = 102.0
```

```
      DO 20 I=1,11
      TIC = TIC + 32.0
      YL(1) = TIC
      YL(2) = TIC
      N = 2 * I - 1
      CALL CHAR(64.0,YL,0,IYNUM(N),4,2,1)
   20 CALL LINE(XL,YL,2,0)
C
C          DRAW Y AXIS LABEL
C
      COORD = 360.0
      DO 30 I=1,8
      COORD = COORD - 20.0
   30 CALL CHAR(30.0,COORD,0,IALTL(I),2,2,1)
      CALL CHAR(30.0,COORD-20.0,0,IMET,3,2,1)
C
C THE FOLLOWING THREE SECTIONS OF CODE PRODUCE AN X AND Y AXIS ON
C THE RIGHT HAND SIDE OF THE PLOT
C
C DRAW THE NEW AXES.
C
      XCD(1) = 300.0
      XCD(2) = 500.0
      XCD(3) = 500.0
      CALL LINE(XCD,YCD,3,0)
C
C DRAW TIC MARKS ON THE RIGHT HAND Y AXIS
C
      TIC = 58.0
      XL(1) = 498.0
      XL(2) = 502.0
      DO 76 I = 1,11
      TIC = TIC + 32.0
      YL(1) = TIC
      YL(2) = TIC
   76 CALL LINE(XL,YL,2,0)
      XL = 460.0
      YL = 410.0
      CALL CHAR(XL,YL,0,IFOUR,4,2,1)
C
C DRAW TICK MARKS ON THE RIGHT HAND X AXIS AND NUMBER THE AXIS
C
      TIC = 275.0
      YL(1) = 88.0
      YL(2) = 92.0
      COORD = TIC - 8.0
      DO 78 K = 1,9
      TIC = TIC + 25.0
      XL(1) = TIC
      XL(2) = TIC
      CALL LINE(XL,YL,2,0)
      COORD = COORD + 25.0
      N = 2 * K - 1
   78 CALL CHAR(COORD,80.0,0,IDIM(N),4,2,1)
C
C DRAW THE RIGHT HAND X AXIS LABLE
C
      CALL CHAR(300.0,70.0,0,ICWD,10,2,1)
C
C  LA... FOR ... DA... ..LE .....
```

```
C
      CALL CHAR(404.0,40.0,0,NAFIL,10,2,1)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C        THIS SECTION LABELS THE MET PLOT FORM AND DRAWS THE CURVES
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C        DETERMINE SOME X AND Y COORDINATES AND TOTAL NUMBER OF POINTS
C        FOR THE CURVES
C
   34 CONTINUE
C
C        THE FOLLOWING IF TEST ALLOWS MET PLOTS TO BE BY-PASSED
C
      IF((MOIN.Y.EQ.1).AND.LVRCRT) GO TO 110
   35 DO 40 I=1,NUM
      IF(ALT(I) .GE. 4000.0)GO TO 50
      CURVEY(I) = ALT(I) * 0.08 + 90.0
      AWDIR(I) = IDIR(I)
   40 APTEMP(I) = PTEMP(I) - 273.16
      I = NUM + 1
   50 ILP = I - 1
C
C        CALL SUBROUTINE TO ROTATE WIND DIRECTION FOR PLOTTING
C
      CALL WINDS(AWDIR,ILP,ISC)
      IF(MFORM.NE.0) GO TO 51
      XAX(1) = XAX(1) * XSCALE
      XINC = (XAX(1) - XAX(2))/12.0
      DO 88 K = 1,10
   88 XWD(K) = XWD(K) * XSCALE
   51 DO 60 I=1,9
      N = 2 * I - 1
      COORD = XWD(N) - 8.0
   60 CALL CHAR(COORD,50.0,0,IWDL(N,ISC),4,2,1)
C
C FAKT IS USED TO SCALE THE DATA---EACH XINC REPRESENTS 5 UNITS
C ON THE X AXIS -- 100 IS WHERE THE AXIS BEGINS --
C
      FAKT = XINC/5.0
      DO 70 I=1,ILP
      WSX(I) = (SPEED(I) * FAKT) + 2*XINC + 100.0
      DTX(I) = (TEMP(I) * FAKT) + 2*XINC + 100.0
      PTX(I) = (APTEMP(I) * FAKT) + 2*XINC + 100.0
      IF(TEMP(I) .LT. -10.0)DTX(I) = 100.0
      IF(TEMP(I) .GT. 50.0)DTX(I) = 460.0 * XSCALE
      IF(APTEMP(I) .LT. -10.0)PTX(I) = 100.0
      IF(APTEMP(I) .GT. 50.0)PTX(I) = 460.0 * XSCALE
   70 WDX(I) = (ABS(AWDIR(I)) * 0.444444 + 300.0) * XSCALE
C
C        WRITE THE DATE AND TIME
C
      CALL CODE
      WRITE (IALPHA,2000) ISDAY,ISMON(1),ISMON(2),ISYEAR
      CALL CHAR(60.,490.,0,IALPHA,11,2,1)
      CALL CODE
      WRITE (IALPHA,2001) ISTIME
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CALL CHAR(204.,490.,0,IALPHA,4,2,1)
      CALL CHAR(246.0,490.0,0,ITIMEZ,1,2,1)
      IF(LVRPLS)CALL CHAR(248.0,490.0,0,LAUNTD(4),2,2,1)
      IF(LVRCRT)CALL CHAR(246.0,490.0,0,LAUNTD(4),2,2,1)
C
C        IF NECESSARY, WRITE THE INSTALLATION
C
      IF(IPLACE .EQ. 0)GO TO 74
      I = IPLACE - IPLACE/3
      IF(IPLACE.EQ.4) I = 2
      CALL CHAR(300.0,490.0,0,INSTAL(1,I),4,2,1)
      XL(1) = 300.0
      XL(2) = 336.0
      YL(1) = 400.0
      YL(2) = 400.0
      CALL LINE(XL,YL,2,0)
C
C        WRITE THE SURFACE PRESSURE, DENSITY, AND STABILIZATION HEIGHT
C
   74 CALL CODE
      WRITE (IALPHA,2002) PRESS(1)
      IF(LVRPLS)CALL CHAR(196.0,475.0,0,IALPHA,6,2,1)
      IF(LVRCRT)CALL CHAR(133.0,475.0,0,IALPHA,6,2,1)
      CALL CODE
      WRITE (IALPHA,2002) SURDEN
      IF(LVRPLS)CALL CHAR(300.0,475.0,0,IALPHA,6,2,1)
      IF(LVRCRT)CALL CHAR(260.0,475.0,0,IALPHA,6,2,1)
      IF(LVRPLS)GO TO 77
      CALL CODE
      WRITE (IALPHA,2002) STBALT
      CALL CHAR(420.0,475.0,0,IALPHA,6,2,1)
      CALL CODE
      WRITE (IALPHA,2003) SIGAZ
      CALL CHAR(416.0,490.0,0,IALPHA,4,2,1)
C
C        WRITE THE DRY TEMPERATURE
C
   77 CALL CODE
      WRITE (IALPHA,2002) TEMP(1)
      CALL CHAR(230.0,450.0,0,IALPHA,6,2,1)
C
C        WRITE THE POTENTIAL TEMPERATURE
C
      A = PTEMP(1) - 273.16
      CALL CODE
      WRITE (IALPHA,2002) A
      CALL CHAR(230.0,440.0,0,IALPHA,6,2,1)
C
C        WRITE THE WIND SPEED
C
      CALL CODE
      WRITE (IALPHA,2002) SPEED(1)
      CALL CHAR(230.0,430.0,0,IALPHA,6,2,1)
C
C        WRITE THE WIND DIRECTION
C
      A = IDIR(1)
      CALL CODE
      WRITE (IALPHA,2002) A
      CALL CHAR(230.0,420.0,0,IALPHA,6,2,1)
```

```
C
C        DRAW THE WIND SPEED LINE
C
C     CALL DLINE(WSX,CURVEY,ILP,0,4,4)
      CALL DLINE(WSX,CURVEY,ILP,0,8,4)
      COORD = CURVEY(ILP) + 3.0
      CALL CHAR(WSX(ILP),COORD,0,ICRVT(1),2,2,1)
C
C        DRAW THE DRY TEMPERATURE LINE
C
      CALL LINE(DTX,CURVEY,ILP,0)
      COORD = CURVEY(ILP) - 5.0
      CALL CHAR(DTX(ILP)+4.0,COORD,0,ICRVT(2),2,2,1)
C
C        DRAW THE POTENTIAL TEMPERATURE LINE
C
      CALL DLINE(PTX,CURVEY,ILP,0,4,4)
      COORD = CURVEY(ILP) + 3.0
      CALL CHAR(PTX(ILP),COORD,0,ICRVT(3),2,2,1)
C
C        DRAW THE WIND DIRECTION LINE
C
      I1 = 1
      DO 80 I=2,ILP
      IF(AWDIR(I) .GE. 0.0)GO TO 80
      NUMP = I - I1
C     CALL DLINE(WDX(I1),CURVEY(I1),NUMP,0,4,8)
      CALL LINE(WDX(I1),CURVEY(I1),NUMP,0)
      I1 = I
   80 CONTINUE
      NUMP = ILP - I1 + 1
C     CALL DLINE(WDX(I1),CURVEY(I1),NUMP,0,4,8)
      CALL LINE(WDX(I1),CURVEY(I1),NUMP,0)
      COORD = CURVEY(ILP) - 5.0
      CALL CHAR(WDX(ILP)+4.0,COORD,0,ICRVT(4),2,2,1)
C
C        DRAW TICK MARKS AT THE VALID DATA POINTS ON THE Y AXIS
C
      XL(1) = 100.0
      XL(2) = 106.0
      DO 90 I=1,ILP
      YL(1) = ALT(I) * 0.08 + 90.0
      YL(2) = YL(1)
   90 CALL LINE(XL,YL,2,0)
C
C        DRAW THE CLOUD
C
      COORD = STRALT * 0.08 + 90.0
C     CALL CLOUD(250.0,COORD)
      CALL CLOUD(400.0,COORD)
C
C        WRITE OUT THE TOP OF THE SURFACE LAYER LINE AND ALLOW IT
C        TO BE MOVED UP AND DOWN
C
      CALL MOVEN(LAYTOP,ILP,2,ITOP,310.0,TSURX,10)
      TOPLAY = ALT(LAYTOP)
C
C        CALCULATE AND WRITE OUT THE EFFECTIVE CLOUD HEIGHT, STBHT
C
      SIGM0 = 0.297674 * STRALT
```

```
      STBHT = STBALT
      CLDRAD = 2.15 * SIGX0
      IV2 = 0
      IF(CLDRAD+STBALT .GE. ALT(LAYTOP))IV2 = 1
      SIGZ0 = SIGX0
      IF(IV2 .EQ. 1)SIGZ0 = (ALT(LAYTOP) - STBALT + CLDRAD)/4.3
      IF(SIGZ0 .GT. 0.0)GO TO 91
      STBHT = 0.5 * ALT(LAYTOP)
      SIGZ0 = 0.64 * STBHT/2.15
      GO TO 92
   91 IF(IV2 .EQ. 1)STBHT = 0.5 * (ALT(LAYTOP) + STBALT - CLDRAD)
   92 WRITE (6,666) STBHT
  666 FORMAT( "STBHT =",E14.6)
C
C DRAW AN ELLIPSE FOR THE EFFECTIVE CLOUD HEIGHT
C --HCEN AND KCEN ARE THE COORDINATES OF THE CENTER OF THE ELLIPSE
C --AAX AND BAX ARE THE MAJOR AND MINOR AXES
C -- THE SIGN OF THE X VALUE IS CHANGED EVERY OTHER TIME THE LOOP
C    IS EXECUTED
C -- THE Y VALUE IS DECREMENTED EVERY OTHER TIME THE LOOP IS EXECUTED
C
      HCEN = 400.0
      KCEN = STBHT * 0.08 + 90.0
      AAX = (GAMMAX*STBALT*0.08)*0.8
      BAX = (0.5 * (GAMMAX*STBALT - STBALT + ALT(LAYTOP)))*0.08
      YINC = BAX/25.0
      SIGN = 1.0
      YPT = KCEN + BAX
C     XPT = HCEN
      YELL(1) = YPT
      XELL(1) = HCEN
      JDEC = 1
      JSW = 1
      DO 99 K=2,100
      IF( JDEC.EQ.1) YPT = YPT - YINC
      RAD = AAX * SQRT(1.0 - ((YPT-KCEN)**2)/(BAX**2))
      XELL(K) = HCEN + SIGN * RAD
      YELL(K) = YPT
      JSW = JSW + 1
      IF( JSW.EQ.2) SIGN = -SIGN
      IF( JSW.EQ.2) JSW = JSW - 2
   99 JDEC = -JDEC
      CALL LINE(XELL,YELL,100,0)
C
C** SPECIFY THE DATA FILE USED.
C
      CALL CHAR(464.0,40.0,0,FILE,6,2,1)
C
C        IF REQUESTED, WRITE OUT THE BOTTOM OF THE SURFACE LAYER
C        LINE AND ALLOW IT TO BE MOVED UP AND DOWN
C
      IF(ICALC .NE. 1)GO TO 100
      CALL MOVEM(LAYBOT,ILP,1,IBOT,414.0,BSURX,5)
      BOTLAY = ALT(LAYBOT)
      CALHT = BOTLAY
C
C        IF NOT RUNNING ON THE PLASMASCOPE, DO NOT ASK HARDCOPY QUESTION
C
  100 IF(LVRCRT)GO TO 110
        . CH     24..      0,1     (1).     1,0)
```

```
      CALL CHARC(168.0,16.0,0,IHARD(10),8,0,0)
      CALL CHARC(232.0,16.0,0,IHARD(14),6,0,0)
      CALL IN(1,JTYPE,0.0,0.0,0,0,0,0,31,0,31,IX,IY)
      IF(IX .GT. 15)GO TO 110
      CALL RWDIS(NAME,1)
      CALL NGRAF
      CALL EXEC(9,RMETQ)
      CALL RWDIS(NAME,0)
C
C        WRITE OUT THE COMMON DISK FILE
C
  110 CALL RWDIS(NAME,1)
C
C        CALL CLEAR AND NGRAF TO CLEAR PLASMASCOPE AND TERMINATE .
C        GRAPHIC MODE
C
CC    CALL CLEAR
      IF(LVRPLS) CALL CLEAR
      CALL NGRAF
C
C
C*** TERMINATE RMTPS ***
      STOP
      END
      END$
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
&RMTPT T=00003 IS ON CR00003 USING 00052 BLKS R=0000

 .FTN4X,L
       SUBROUTINE RWDIS(NAME,JJ),                         *REEDS,CAB810815
 C
 C  ------------------------------------------------------------
 C  -                                                          -
 C  -    THIS SUBROUTINE READS AND WRITES COMMON AREAS TO DISC -
 C  -    TO BE UTILIZED BY THE VARIOUS REED PROGRAMS.          -
 C  -                                                          -
 C  ------------------------------------------------------------
 C
 C  >>>   COMMON AREAS
 C
 C
 C  >>>   MATH PARAMETERS
       COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
 C
 C  >>>   VEHICLE PARAMETERS
       COMMON VPAR(18),SIGHCL
 C
 C  >>>   OPTION PARAMETERS
       COMMON YPOS,YES,ISNDFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
      .       NUMRUN,LU,NUM,MONLY,MFORM,LUPRT,ITDU,IPL1,IPL2,IPL3
       INTEGER YES
 C
 C  >>>   MET DATA
       COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
      .       SURDEN,FILE(3)
       INTEGER FILE
 C
 C  >>>   CALCULATION TIME
       COMMON ITIME,IDAY,IMONTH(2),IYEAR
 C
 C  >>>   SOUNDING TIME
       COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
 C
 C  >>>   LAUNCH TIME
       COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
       LOGICAL LTIME
 C
 C  >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
       COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
      .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
 C
 C  >>>   DIFFUSION COEFFICIENTS
       COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
      .       CLDLNG,AVSPD,AVDIR,SIGAZ
 C
 C  >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
       COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
 C
 C  >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
       EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
      .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
      .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
      .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
      .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
      .            (F[illegible]3,V[illegible](16),[illegible]
       INTEGER [illegible]CB(144),[illegible](557)
```

```
      DIMENSION NAME(3) , ISIZE(2)
      EQUIVALENCE (OBUF(1),FI)

CBJ *** ISIZE(1) = # BLOCKS EXPECTED TO BE WRITTER READ DURING RUN
CBJ *** ISIZE(2) = # WORDS IN THE COMMON BLOCK TO BE TRANSFERRED
      DATA ISIZE/30,557/

CBJ *** IF THE SCRATCH DATA FILE (i.e. *REED2) ISN'T THERE, CREATE IT
CBJ *** B E W A R E !!  BE SURE FILE IS PURGED BEFORE STARTING RUN !!
      CALL OPEN(O  B,IERR,NAME,0)
      IF( (IERR.LT.0) .AND. (IERR.NE.-6) ) WRITE(6,778) IERR
  778 FORMAT(" *** ATTEMPT TO OPEN, IERR =",I4)
      IF(IERR.LT.0)
     .   CALL CREAT(ODCB,IERR,NAME,ISIZE,3,0,0,144)
      IF(IERR.LE.0) WRITE(6,799) IERR
  799 FORMAT(" *** CREATE/OPEN, IERR=",I4)

CBJ *** READ OR WRITE, AS PER ORDER
      IF(JJ.EQ.1)CALL WRITF(ODCB,IERR,OBUF,ISIZE(2))
      IF(JJ.EQ.0)CALL READF(ODCB,IERR,OBUF,ISIZE(2))
      CALL CLOSE(ODCB,IERR)
      RETURN
      END
      SUBROUTINE MOVEM(JND,MAXJND,MINJND,LAB,XLABEL,XLINE,NLINE),
     .MOVEM                  *CAB,010813
C
C -------------------------------------------------------------------
C -                                                                 -
C -  THIS SUBROUTINE MOVES THE TOP AND BOTTOM SURFACE LAYER LINES   -
C -  UP AND DOWN ON THE METEROLOGICAL PROFILE PLOTS, AS SPECIFIED   -
C -  BY THE USER.                                                   -
C -                                                                 -
C -------------------------------------------------------------------
C
C  >>>   COMMON AREAS
C
C
C  >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>   VEHICLE PARAMETERS
      COMMON VPAR(10),SIGHCL
C
C  >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNDFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,NONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C  >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C  >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C  >>>   LAUNCH TIME
```

```
      COMMON LTIME,LTIN,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME, LVERSN
C
C  >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C  >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C  >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C  >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
C*** FORMATS
91    FORMAT(A2)
99    FORMAT(2X," UP/DOWN/CONT. SECTION ****IANS= ",A2,"  IX,IY=",2I3)
 2000 FORMAT (F6.1)
 2001 FORMAT (" "I3",0")
      INTEGER QUES(13),ANS1,ANS2(2),ANS3(4),BLANKS(26)
      DIMENSION LAB(1),XLINE(1),YLINE(2),JNDALT(3),JNDVAR(3,4)
      EQUIVALENCE (JNDVR1,JNDVAR(1,1)),(JNDVR2,JNDVAR(1,2)),
     .            (JNDVR3,JNDVAR(1,3)),(JNDVR4,JNDVAR(1,4))
      DATA QUES/2HMO,2HVE,2H  ,2H  ,2H O,2HF ,2HSU,2HRF,2HAC,2HE ,2HLA,
     .          2HYE,2HR:/
      DATA ANS1/2HUP/, ANS2/2HDO,2HWN/, ANS3/2HCO,2HNT,2HIN,2HUE/
      DATA BLANKS/26*2H  /
C
C*** SET LU FLAG
      CALL VERSN(I)
      LVERSN = I .EQ. 7
C
C*** INITIALIZE NEW ALTITUDE LEVEL INDEX
      NEWJND = 0
C
C*** START LOOP TO FIND DESIRED LEVEL
    1 YLINE(1) = ALT(JND) * 0.08 + 90.0
      YLINE(2) = YLINE(1)
C
C*** DRAW LAYER
      DO 4 I=1,NLINE
      J = 2 * I - 1
    4 CALL LINE(XLINE(J),YLINE,2,0)
C
C*** IF FIRST PASS, WRITE QUESTION.. OTHERWISE SKIP TO ANSWER
      IF(NEWJND .EQ. JND)GO TO 11
      QUES(3) = LAB(1)
      QUES(4) = LAB(2)
CC
CC***ATTEMPT TO PREVENT OVERWRITING ON CRT VERSION...JSH

      - -YPOS - 93.0
```

```
      IF(LVERSH) Y = 1.0
CC    CALL CHARC(0.0,1.0,-1,QUES,26,3,0)
      CALL CHARC(0.0,Y,-1,QUES,26,3,0)
CC    CALL CHARC(29.0,1.0,-1,ANS1,2,0,0)
      CALL CHARC(29.0,Y,-1,ANS1,2,0,0)
CC    CALL CHARC(36.0,1.0,-1,ANS2,4,0,0)
      CALL CHARC(36.0,Y,-1,ANS2,4,0,0)
CC    CALL CHARC(43.0,1.0,-1,ANS3,8,3,0)
      CALL CHARC(43.0,Y,-1,ANS3,8,3,0)
11    CONTINUE
C
C*** THIS IS THE UP/DOWN/CONTINUE DECISION AREA
C
C*** IF PLASMASCOPE, GO TO TOUCH INPUT
      IF(LVERSH) GO TO 12
C
C*** DEFAULT IS UP
      IX = 10
      READ(LU,91)IANS
      IF( (IANS.NE.2HDD) .AND. (IANS.NE.2HCO) ) GO TO 13
C
C*** THIS IS CONTINUE OR DOWN
      IX = 25
      IF( IANS.NE.2HDO ) GO TO 13
C
C*** THIS IS DOWN
      IX = 10
      GO TO 13
12    CONTINUE
      CALL IN(1,J,0.0,0.0,0,0,0,0.31,0.31,IX,IY)
13    CONTINUE
C
C*** IF NOT CONTINUE, MOVE THE LINE--BUT DO NOT LABLE IT OR FILL IN THE TABLE
C     WRITE(6,99) IANS,IX,IY
      IF(IX .LE. 20)GO TO 25
C
C*** IF CONTINUE, LABLE THE LINE AND FILL IN THE TABLE
      Y = YLINE(1) + 2.0
      CALL CHAR(460.0,Y,0,LAB,4,2,1)
      Y = Y - 10.0
      CALL CODE
      WRITE (JHDALT,2000) ALT(JHD)
      CALL CHAR(460.0,Y,0,JHDALT,6,2,1)
      CALL CODE
      WRITE (JHDVR1,2000) TEMP(JHD)
      YLABEL = PTEMP(JHD) - 273.16
      CALL CODE
      WRITE (JHDVR2,2000) YLABEL
      CALL CODE
      WRITE (JHDVR3,2000) SPEED(JHD)
      CALL CODE
      WRITE (JHDVR4,2001) IDIR(JHD)
      YLABEL = 450.0
      DO 14 I=1,4
      CALL CHAR(XLABEL,YLABEL,0,JHDVAR(1,I),6,2,1)
   14 YLABEL = YLABEL - 10.0
C  22 IF(NEWJHD .EQ. JHD)GO TO 11
C     DO 24 I=1,NLINE
C     J = 2 * I - 1
C  24 CALL LINE(XLINE(J),YLINE,2,1)
```

```
      CALL CHAR(460.0,Y,0,JHDALT,6,1,1)
      Y = Y + 10.0
      CALL CHAR(460.0,Y,0,LAB,4,1,1)
      YLABEL = 450.0
      DO 26 I=1,4
      CALL CHAR(XLABEL,YLABEL,0,JHDVAR(1,I),6,1,1)
   26 YLABEL = YLABEL - 10.0
      CALL CHAR(0.0,1.0,-1,BLANKS,51,0,0)
      RETURN
   25 IF(IX .GE. 17)GO TO 28
      NEWJHD = MIN0(JHD + 1,MAXJHD)
      JHD = NEWJHD
      GO TO 1
   28 NEWJHD = MAX0(JHD - 1,MINJHD)
      JHD = NEWJHD
      GO TO 1
      END
      SUBROUTINE WINDS(WD,NWD,ISC)
C
C    -------------------------------------------------------
C    -                                                     -
C    -  THIS SUBROUTINE COMPUTES THE WIND DIRECTION LABELS  -
C    -  FOR PLOTTING                                       -
C    -                                                     -
C    -------------------------------------------------------
C
      DIMENSION WD(1),ENDPT(4),NUMUP(4)
      EQUIVALENCE (J,LEAST)
      DATA ENDPT/0.0,90.0,180.0,270.0/
      DO 2 I=1,4
    2 NUMUP(I) = 0
      WD2 = WD(1)
      DO 8 I=2,NWD
      WD1 = WD2
      WD2 = WD(I)
      DO 6 J=1,4
      C1 = WD1 - ENDPT(J)
      IF(C1 .LT. 0.0)C1 = C1 + 360.0
      C2 = WD2 - ENDPT(J)
      IF(C2 .LT. 0.0)C2 = C2 + 360.0
      IF(ABS(C1-C2) .LE. 180.0)GO TO 6
      NUMUP(J) = NUMUP(J) + 1
    6 CONTINUE
    8 CONTINUE
      ISC = 1
      LEAST = NUMUP(1)
      DO 12 I=2,4
      IF(NUMUP(I) .GE. LEAST)GO TO 12
      ISC = I
      LEAST = NUMUP(I)
   12 CONTINUE
      DO 17 I=1,NWD
      WD(I) = WD(I) - ENDPT(ISC)
      IF(WD(I) .LT. 0.0)WD(I) = WD(I) + 360.0
   17 CONTINUE
      WD2 = WD(1)
      DO 22 I=2,NWD
      WD1 = WD2
      WD2 = WD(I)
      [illegible]
```

```
      WD(I) = - WD(I)
   22 CONTINUE
      RETURN
      END
      SUBROUTINE CLOUD(XP,YP)
C
C   ---------------------------------------------------------
C   -                                                       -
C   - THIS SUBROUTINE DRAWS A CIRCULAR CLOUD FOR THE MET    -
C   - PROFILE, AT A SPECIFIED X AND Y POSITION.             -
C   -                                                       -
C   ---------------------------------------------------------
C
C   >>>   COMMON AREAS
C
C
C   >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C   >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C   >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNOFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTHP,
     .       NUMRUN,LU,NUM,MONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C   >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C   >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C   >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C   >>>   LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME
C
C   >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C   >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C   >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C   >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .               (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .               (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .               (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .               (PHCL,VPAR(13)),(PCO,VPAR(14)),([illegible],VPAR(15)),
     .               (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DIMENSION X(61),Y(61)
C
C THE MULTIPLICATIVE CONSTANT .8 WAS INTRODUCED TO DRAW A CIRCULAR
C CLOUD WITH XSCALE IN USE.
C
      RADIUS = GAMMAX * STBALT * 0.08
      DO 7 I=1,61
      X(I) = (RADIUS * COS(0.01745329252 * FLOAT(6 * I))) * .8 + XP
    7 Y(I) = RADIUS * SIN(0.01745329252 * FLOAT(6 * I)) + YP
      CALL LINE(X,Y,61,0)
      RADIUS = 5.0
      X(1) = XP + RADIUS
      X(2) = XP
      X(3) = XP - RADIUS
      X(4) = XP
      X(5) = X(1)
      Y(1) = YP
      Y(2) = YP + RADIUS
      Y(3) = YP
      Y(4) = YP - RADIUS
      Y(5) = Y(1)
      CALL LINE(X,Y,5,0)
      X(2) = XP - RADIUS
      Y(2) = YP
      CALL LINE(X,Y,2,0)
      X(3) = XP
      Y(3) = YP + RADIUS
      CALL LINE(X(3),Y(3),2,0)
      RETURN
      END
      END$
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
&RCONS T=00003 IS ON CR00003 USING 00073 BLKS R=0000

FTN4X,L
      PROGRAM RCONS(3),REEDS CONC. & DOSAGE PROG...UPDATED 12 22 79, CBJ
C
C     *****************************************************************
C     *                                                               *
C     *     CONCENTRATION AND DOSAGE PROGRAM -- A PROGRAM OF THE      *
C     *     REEDS SERIES OF PROGRAMS                                  *
C     *                                                               *
C     *****************************************************************
C
C
C  >>>    COMMON AREAS
C
C
C  >>>    MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>    VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C  >>>    OPTION PARAMETERS
      COMMON YPOS,YES,ISNDFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,MONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>    MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C  >>>    CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C  >>>    SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C  >>>    LAUNCH TIME
      COMMON LTIME,LTIN,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME, LVERSN
C
C  >>>    CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C  >>>    DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C  >>>    CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C  >>>    EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATH,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),[illegible]VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PH0,VPAR(17)),(GAMMAX,VPAR(18))
```

```
C
C         OUTPUT FORMAT STATEMENTS
C
 90     FORMAT(A1)
  200 FORMAT ("1"12X"CLOUD CONCENTRATIONS AND DOSAGES"/
     .         "0DISTANCE"4X"CONCENTRATION"5X"DOSAGE"6X
     .          "TIME AFTER LAUNCH(SEC)",11X,"POSITION FROM PAD"/
     .          " (METERS)"8X"(PPM)"8X"(PPM SEC)"8X"START"3X"FINISH",
     .          "                    RANGE     AZ       ")
  201 FORMAT (1X  .1,8XF7.3,8XF7.3,9XF5.1,3XF5.1,16X,2F8.1)
  202 FORMAT (//"0****POINT OF MAXIMUM CONCENTRATION****"/
     .          6X"RANGE FROM PAD(M): "F8.1/
     .          6X"DIRECTION(DEG): "F5.1/
     .          6X"HEIGHT(M): "F6.1/
     .          6X"MAXIMUM CONCENTRATION(PPM): "F6.3)
  203 FORMAT (//"0****POINT OF MAXIMUM DOSAGE****"/
     .          6X"RANGE FROM PAD(M): "F8.1/
     .          6X"DIRECTION(DEG): "F5.1/
     .          6X"HEIGHT(M): "F6.1/
     .          6X"MAXIMUM DOSAGE (PPM SEC): "F6.3)
  204 FORMAT (//"0****CONCENTRATIONS AND DOSAGES WITH 10 DEGREE "
     .          "UNCERTAINTIES****")
  205 FORMAT ("0"5X"RANGE(M): "F7.1/
     .          6X"AZIMUTH(DEG): "F5.1/
     .          6X"MATERIAL"5X"CONCENTRATION(PPM)"11X"DOSAGE(PPM)")
  206 FORMAT (4I5,I2)
C 207 FORMAT (7X3A2,6XF8.3" +/- "F8.3,4XF8.3" +/- "F8.3)
  207 FORMAT (7X3A2,6XE8.3" +/- "F8.3,4XF8.3" +/- "F8.3)
C
C         TYPE AND DIMENSION STATEMENTS
C
      INTEGER RISOP(3)
      DIMENSION FACT(3),CMNPL(3),DMNPL(3),MATS(3,5),NAME(3),
     .          NAMEF(3),ILINE(32),IDATAF(10),IERS(32),
     .          XDISTV(81),DOSV(81),CONCV(81)
C
C         DATA STATEMENTS
      DATA NAME/2H=R,2HEE,2HD2/,NAMEF/2H?R,2HCO,2HNS/
      DATA IERS/32*2H  /
C
      DATA FACT/0.0,-0.174533,0.174533/
      DATA MATS/2H  ,2HHC,2HL ,2H  ,2H C,2HO ,
     .          2H  ,2HCO,2H2 ,2H A,2HL2,2HO3,
     .          2H  ,2H N,2HO /
      DATA RISOP/2HRI,2HSO,1HS/
C
C         CALL GRAF TO INITIALIZE SCOPE (APPROPRIATE ONLY WHEN USING
C         PLASMASCOPE)
C
      CALL GRAF(1)
C
C         FIND OUT IF THIS IS THE CRT OR PLASMASCOPE VERSION OF REED2
C         BEING RUN
C
      CALL VERSN(I)
      LVERSN = I .EQ. 0
C
C         READ COMMON DISK FILE =REED
C
      CALL RWDISK(NAME,0)
```

```
C
C         DO LOOP FOR CONCENTRATION AND DOSAGE CALCULATIONS
C
C         XDIST - RANGE FROM STABILIZATION
C         DOSPK - DOSAGE
C         DOSMAX - MAXIMUM DOSAGE
C         CONCPK - CONCENTRATION
C         CONMAX - MAXIMUM CONCENTRATION
C
      NUMV = 0
      YES = 1HY
      NO = 1HN
      CONMAX = 0.0
      DOSMAX = 0.0
      ACTVOL = PI43 * CLDRAD * CLDRAD * CLDRAD
      TOTVOL = ACTVOL
      IF(IV2 .EQ. 1)ACTVOL = PI * (ALT(LAYTOP) + CLDRAD - STBALT)**2 *
     .                        (2.0 * CLDRAD - ALT(LAYTOP) + STBALT)/3.0
      SIGHCL = SIGHCL * ACTVOL/TOTVOL
C
C*** PREPARE CONSTANTS FOR RELATIVE PAD/CENTERLINE CALCULATIONS
      STBAZR = DEGRAD * STBAZD
      COSSTB = COS(PI - STBAZR)
      SINSTB = SIN(PI - STBAZR)
      XSTAB = STBRNG * COSSTB
      YSTAB = STBRNG * SINSTB
      AVDIRR = DEGRAD * AVDIR
      COSAV = COS(PI - AVDIRR)
      SINAV = SIN(PI - AVDIRR)
      WRITE (LUPRNT,200)
C
C*** SET OFF-CENTERLINE DISTANCE TO 0 FOR CENTERLINE CALCULATION
      YDIST = 0.0
      DO 59 I=0,20000,250
C
      NUMV = NUMV + 1
      XDIST = I
C
C*** GET PAD-RELATIVE RANGE AND AZIMUTH TO APPROPRIATE CENTERLINE POSITION
      XREL = XSTAB + XDIST*COSAV
      YREL = YSTAB + XDIST*SINAV
      SITRNG = SQRT(XREL*XREL + YREL*YREL)
      DIRPCL = ATAN2(YREL,XREL)
      SITAZD = RADDEG * (PI - DIRPCL)
      XDISTV(NUMV) = XDIST
C
      CALL DFEXP(LAYTOP,1000.0)
C
C         THE WRITE STATEMENTS BELOW HAVE BEEN ADDED AS TEMPORARY
C         AIDS IN VALIDATING CALCULATIONS FOR DOSAGE AND CONCENTRATION.
C
      DOSPK = SIGHCL * EXPZ/(TWOPI * SIGY * AVSPD * SQRT(0.5 * SQSIGZ))
      DOSV(NUMV) = DOSPK
      CONCPK = DOSPK * AVSPD/(SQR2PI * SIGX)
      CONCV(NUMV) = CONCPK
C
      IF(DOSPK .LE. DOSMAX)GO TO 55
      RNGSMD = XDIST
      DOSMAX = DOSPK
C
```

```
   55 IF(CONCPK .LE. CONMAX)GO TO 58
      RNGSMC = XDIST
      CONMAX = CONCPK
      SGXMAX = SIGX
      SGYMAX = SIGY
C
   58 IF(AMOD(XDIST,1000.0) .NE. 0.0)GO TO 59
C
      ARVTIM = STBTIM + (XDIST - CLDLNG)/AVSPD
      DEPTIM = STBTIM + (XDIST + CLDLNG)/AVSPD
      WRITE (LUPRNT,201) XDIST,CONCPK,DOSPK,ARVTIM,DEPTIM,
     .                   SITRNG, SITAZD
C
   59 CONTINUE
C
C        IF THIS IS A PRODUCTION RUN, SKIP THE PLOTTING
C
      IF(IRUN .EQ. 3)GO TO 61
C
C        IF REQUESTED, PLOT THE CENTERLINE DOSAGE AND CONCENTRATION
C        VALUES
C
      CALL DREAD(NAMEF,2,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.,YPOS,0,ILINE,42,3,0)
      CALL CHAR(304.,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.,YPOS,0,ILINE(30),6,0,0)
C
C         IF PLASMASCOPE VERSION, SKIP TO THE CALL TO "IN".
C
      IF(LVERSN) GO TO 222
C
C         OTHERWISE, RESPOND WITH Y FOR YES, N FOR NO TO ANSWER
C         QUESTIONS.
C
      READ(LU,90) IANS
      IF(IANS.EQ.1HO)STOP
C
C         "YES" IS THE DEFAULT CASE
C
      IX = 25
C
C         THIS IS THE "NO" RESPONSE
C
      IF(IANS.NE.NO)GO TO 223
      IX = 30
      GO TO 223
  222 CONTINUE
      CALL IN(1,JTYPE,0.,0.,0,0,0,31,0,31,IX,IY)
  223 CONTINUE
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)CALL CHAR(464.,YPOS,0,IERS,6,0,0)
      IF(IX .GT. 25)CALL CHAR(304.,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.
      IF(IX .GT. 25)GO TO 61
      EX = ALOCT(DOSMAX)
      IEXPD = EX
      IF(EX .LT. 0.0)IEXPD = IEXPD - 1
      IEXPD = - IEXPD
      [illegible] = ALO[illegible](CONMAX)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IEXPC = EX
      IF(EX .LT. 0.0)IEXPC = IEXPC - 1
      IEXPC = - IEXPC
      CALL CPLOT(XDISTV,DOSV,CONCV,NUMV,IEXPD,IEXPC)
      CALL LABEL(IEXPD,IEXPC)
C
C        CALCULATE AND WRITE OUT THE POINT OF MAXIMUM CONCENTRATION
C
 61   CONTINUE
      XDIST = RNGSMC * COSAV + XSTAB
      YDIST = RNGSMC * SINAV + YSTAB
      RNGPMC = SQRT(XDIST * XDIST + YDIST * YDIST)
      ARG1 =  ATAN2(YDIST,XDIST)
      DIRPMC = PI - ARG1
      DIRPMC = RADDEG * DIRPMC
      WRITE (LUPRNT,202) RNGPMC,DIRPMC,CALHT,CONMAX
C
C        CALCULATE AND WRITE OUT THE POINT OF MAXIMUM DOSAGE
C
      XDIST = RNGSMD * COSAV + STBRNG * COSSTB
      YDIST = RNGSMD * SINAV + STBRNG * SINSTB
      RNGPMD = SQRT(XDIST * XDIST + YDIST * YDIST)
      ARG1 =  ATAN2(YDIST,XDIST)
      DIRPMD = PI - ARG1
      DIRPMD = RADDEG * DIRPMD
      WRITE (LUPRNT,203) RNGPMD,DIRPMD,CALHT,DOSMAX
C
C        IF THIS IS A PRODUCTION RUN, SKIP THE OFF CENTER CONCENTATION
C        SECTION AND THE CALL OF PROGRAM RISOP -- IF NOT, DO OFF CENTER
C        CONCENTRATIONS IF DESIRED
C
      IF(IRUN .EQ. 3)GO TO 88
C
C        ARE OFF CENTER CONCENTRATIONS DESIRED?
C
      CALL DREAD(NAMEF,3,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.,YPOS,0,ILINE,30,3,0)
      CALL CHAR(384.0,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.0,YPOS,0,ILINE(30),6,0,0)
C
C        IF PLASMASCOPE VERSION, SKIP TO THE CALL TO "IN".
C
      IF(LVERSN) GO TO 612
C
C        OTHERWISE, RESPOND WITH Y FOR YES, N FOR NO TO ANSWER
C        QUESTIONS.
C
      READ(LU,90) IANS
      IF(IANS.EQ.1H0)STOP
C
C        "YES" IS THE DEFAULT CASE
C
      IX = 25
C
C        THIS IS THE "NO" RESPONSE
C
      IF(IANS .NE. NO) GO TO 613
      IX = 30
      GO TO 613
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
612   CONTINUE
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
613   CONTINUE
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)CALL CHAR(464.,YPOS,0,IERS,6,0,0)
      IF(IX .GT. 25)CALL CHAR(364.,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.
      IF(IX .GT. 25)GO TO 81
C
C        OFF CENTER CONCENTRATIONS ARE DESIRED
C
      WRITE (LUPRNT,204)
C
      CALL ORGIN(IXSET,IYSET,LVERSH)
C
      ITER = 0
C
C        LOOP ON OFF CENTER CONCENTRATION REQUESTS
C
      CALL DREAD(NAMEF,5,ILINE)
      CALL LERS(YPOS,LVERSH)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 16.
      IFLG = 0
   71 ITER = ITER + 1
      IF(IRUN .NE. 1) GO TO 713
C
C        THE FOLLOWING CALCULATIONS AND READ AND WRITE SATEMENTS ARE
C        A TEMPORARY AID IN VERIFYING THE CONCENTRATION AND ISOPLETH
C        PLOTS.
C
      WRITE(LU,711)
711   FORMAT(1X," ENTER A VALUE FOR XDIST? _")
      READ(LU,*) XDIST
      DPC=SQRT(STBRNG**2 +XDIST**2 - 2.*STBRNG*XDIST*COS(PI+STBAZR
     .   -AVDIRR))
      B = PI - ABS(AVDIRR - STBAZR)
      DPCAZ=RADDEG * ATAN((XDIST * SIN( B ) * 2. * STBRNG)
     .      /(STBRNG**2+DPC**2-XDIST**2))
      FLIP = 1.0
      IF(AVDIR .LE. STBAZD) FLIP = -1.
      DPCAZ = FLIP*ABS(DPCAZ) + STBAZD
      WRITE(LU,712) DPC,DPCAZ
      WRITE (6,371) XDIST,DPC,DPCAZ
  371 FORMAT(" DISTANCE AND AZIMUTH FOR XDIST",F6.1,"IS",2F9.2)
712   FORMAT(1X," DISTANCE AND AZIMUTH FROM PAD IS:" F10.3,3X,F10.3)
713   CONTINUE
C
C        READ IN AND WRITE OUT THE RANGE AND AZIMUTH FOR THE
C        OFF CENTER CONCENTRATION CALCULATION -- ENTERING A RANGE OF 0
C        TERMINATES THE PROCEDURE
C
CC
CC***THE WRITE STATEMENTS WITH FORMATS 351 AND 360 HAVE BEEN ADDED
CC***TO ADDRESS THE CURSOR FOR ANSWERING THE QUESTION FOR RANGE
CC***AND AZIMUTH ON THE CRT ONLY...JSH
CC
CC    IF(YPOS.LT.48.) YPOS = 458.
      IF(YPOS .LT. 48.0 .AND. LVERSH) YPOS = 458
      [illegible]
```

```
      YPOS = YPOS - 16.
      IF(LVERSH) YPOS = YPOS - 16.
      IF(YPOS .LT. 48.0.AND.LVERSH) YPOS = 458
      IF(LVERSH) CALL LERS(YPOS,LVERSH)
      CALL CHAR(0.0,YPOS,0,ILINE,64,0,0)
      IF(LVERSH)GO TO 356
      JXAX = 14
      IPOS = -2
      WRITE(LU,351)JXAX,IPOS
 351  FORMAT(" {&a"I2"c"I3"R{")
 356  NIN = 7
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,112.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CODE
      READ (IDATAF,*) SITRNG
      IF(SITRNG .LE. 0.0)GO TO 78
      IF(LVERSH)GO TO 365
      JXAX = 34
      WRITE(LU,360)JXAX,IPOS
 360  FORMAT(" {&a"I2"c"I3"R{")
 365  NIN = 7
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,272.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CODE
      READ (IDATAF,*) SITAZD
      IF(YPOS.LT.0.0)GO TO 380
      YPOS = YPOS - 16.
CC    IF(YPOS .LT. 48.0)YPOS = 458.0
      IF(YPOS .LT. 48.0.AND.LVERSH)YPOS = 458.0
 380  WRITE (LUPRNT,205) SITRNG,SITAZD
C
 73   SITAZR = DEGRAD * SITAZD
      COSSIT = COS(PI - SITAZR)
      SINSIT = SIN(PI - SITAZR)
      XSITEP = SITRNG * COSSIT
      YSITEP = SITRNG * SINSIT
C
C        ON THE PLOTTER, WRITE OUT AN ASTERISK AND THE ITERATION
C        NUMBER AT THE LOCATION WHERE THE OFF CENTER CONCENTRATION
C        CALCULATION IS DESIRED
C
      IXPLOT = IXSET + 0.2631 * XSITEP
      IYPLOT = IYSET + 0.3545 * YSITEP
      WRITE (IPL3) -1,1,IXPLOT,IYPLOT
      CALL SYMBL(100,IPL25,1H*,IPL3)
      IXPLOT = IXPLOT + 75
      WRITE (IPL3) -1,1,IXPLOT,IYPLOT
      WRITE (IPL3,206) 100,0,0,125,ITER
C
C        CALCULATE THE CONCENTRATIONS AND DOSAGES AT THIS POINT PLUS
C        10 DEGREES UNCERTAINTIES ON EITHER SIDE
C
C*** CALCULATE DISTANCE/ANGLE FROM STABILIZATION TO SITE OF INTEREST
      XDEL = XSITEP - XSTAB
      YDEL = YSITEP - YSTAB
      RDEL = SQRT(XDEL*XDEL + YDEL*YDEL)
      BETAF = PI - AVDIRR  - ATAN2(YDEL,XDEL)
      CHECK = ATAN2(YDEL,XDEL)
      BET = RADDEG * BETAF
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DO 74 I=1,3
      ARGI = BETAF - FACT(I)
C
C*** CALCULATE DISTANCE FROM CLOUD CENTER TO SITE OF INTEREST
      YCCSIT =   RDEL * SIN(ARGI)
      XCCSIT = RDEL * COS(ARGI)
      XDIST = XCCSIT
      CALL DFEXP(LAYTOP,1000.0)
C
C*** EXP(- Y * Y   IS NOT DEFINED...YCCSIT IS BEING SUB'D FOR Y .
C*** Y IS PRINTED JUST TO PROVE THE CASE
C     WRITE(6,997)BET,XCCSIT,YCCSIT,XDIST,YDIST
997   FORMAT(//"  BET, XCCSIT, YCCSIT, XDIST, YDIST= ",5F8.1)
      DOS = SIGHCL * EXPZ * EXP(-YCCSIT *YCCSIT/(2.0 * SIGY * SIGY[illegible]
     .      (TWOPI * SIGY * AVSFD * SQRT(0.5 * SQSIGZ))
      CONC = DOS * AVSFD/(SQR2PI * SIGX)
      CMNPL(I) = CONC
C     WRITE(LUPRNT,991)AVDIR,AVDIRR,BETAF,ARGI,XDEL,YDEL,XDIST,YDIST,
C                          XSITEP, YSITEP, XSTAB, YSTAB
991   FORMAT(/5X"AVDIR",4X"AVDIRR",5X"BETAF",6X"ARGI",6X"XDEL",
     .         6X"YDEL",5X"XDIST", 5X"YDIST",4X"XSITEP",4X"YSITEP",
     .         5X"XSTAB",5X"YSTAB"
     .      /12F10.1)
   74 DMNPL(I) = DOS
C
C        CALCULATE AND WRITE OUT THE CONCENTRATION AND DOSAGE FOR
C        EACH MATERIAL
C
      DELC = ABS(0.5 * (2.0 * CMNPL(1) - CMNPL(2) - CMNPL(3)))
      DELD = ABS(0.5 * (2.0 * DMNPL(1) - DMNPL(2) - DMNPL(3)))
      WRITE (LUPRNT,207) (MATS(I,1),I=1,3),CMNPL(1),DELC,DMNPL(1),DELD
C
      ARGI = PCO/PHCL
      CONC = ARGI * CMNPL(1)
      DLC = ARGI * DELC
      DOS = ARGI * DMNPL(1)
      DLD = ARGI * DELD
      WRITE (LUPRNT,207) (MATS(I,2),I=1,3),CONC,DLC,DOS,DLD
C
      ARGI = PCO2/PHCL
      CONC = ARGI * CMNPL(1)
      DLC = ARGI * DELC
      DOS = ARGI * DMNPL(1)
      DLD = ARGI * DELD
      WRITE (LUPRNT,207) (MATS(I,3),I=1,3),CONC,DLC,DOS,DLD
C
      ARGI = PAL2O3/PHCL * 0.43882420 * PRESS(1)/
     .                            (TEMP(1) + 273.16)
      CONC = ARGI * CMNPL(1)
      DLC = ARGI * DELC
      DOS = ARGI * DMNPL(1)
      DLD = ARGI * DELD
      WRITE (LUPRNT,207) (MATS(I,4),I=1,3),CONC,DLC,DOS,DLD
C
      ARGI = PHO/PHCL
      CONC = ARGI * CMNPL(1)
      DLC = ARGI * DELC
      DOS = ARGI * DMNPL(1)
      DLD = ARGI * DELD
      WRITE (LUPRNT,207) (MATS(I,5),I=1,3),CONC,DLC,DOS,DLD
```

```
C
C        REQUEST ANOTHER POINT FOR AN OFF CENTER CONCENTRATION
C        CALCULATION
C
   77 GO TO 71
C
C        FINISHED WITH OFF CENTER CONCENTRATIONS
C
   78 YPOS = YPOS - 32.0
C
C        IS AN ISOPLETH PLOT DESIRED?
C
   81 CALL DREAD(NAMEF,4,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.,YPOS,0,ILINE,62,3,0)
      CALL CHAR(384.,YPOS,0,ILINE(35),8,3,0)
      CALL CHAR(464.,YPOS,0,ILINE(30),6,0,0)
C
C        IF PLASMASCOPE VERSION, SKIP TO THE CALL TO "IN"
C
      IF(LVERSN) GO TO 812
C
C        OTHERWISE RESPOND WITH Y FOR YES AND N FOR NO TO ANSWER
C        QUESTIONS.
C
      READ(LU,90) IANS
      IF(IANS.EQ.1HN)STOP
C
C        "YES" IS THE DEFAULT CASE
C
      IX = 25
C
C        THIS IS THE "NO" RESPONSE
C
      IF(IANS.NE.NO) GO TO 813
      IX = 30
      GO TO 813
  812 CONTINUE
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
  813 CONTINUE
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)CALL CHAR(464.,YPOS,0,IERS,6,0,0)
      IF(IX .GT. 25)CALL CHAR(384.,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.
C
C        IF AN ISOPLETH PLOT IS DESIRED, CALL THE PROGRAM RISOP
C
      IF(IX .GE. 28)GO TO 87
      CALL NGRAF
      CALL RWDIS(NAME,1)
      CALL EXEC(9,RISOP)
      CALL RWDIS(NAME,0)
      CALL GRAF(1)
   87 CONTINUE
C
C        CALL NGRAF TO RETURN SCOPE TO NORMAL MODE OF OPERATION
C        (APPROPRIATE ONLY WHEN USING PLASMASCOPE)
C
   80 CALL NGRAF
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C         WRITE OUT THE COMMON DISK FILE
C
      CALL RUDIS(NAME,1)
C
C         RETURN TO THE MAIN PROGRAM REEDS
C
      STOP
C
     .END
      END$
```

ORIGINAL PAGE IS
OF POOR QUALITY

&RCONT T=00003 IS ON CR00003 USING 00092 BLKS R=0000

```
FTN4,L
      SUBROUTINE RWDIS(NAME,JJ) , ...UPDATED 12 22 79, C. BJORK (CBJ)
C
C  --------------------------------------------------------------
C  -                                                            -
C  -   THIS SUBROUTINE READS AND WRITES COMMON AREAS TO DISC    -
C  -   TO BE UTILIZED BY THE VARIOUS REED PROGRAMS.             -
C  -                                                            -
C  --------------------------------------------------------------
C
C  >>>   COMMON AREAS
C
C
C  >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C  >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNDFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,MONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C  >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C  >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C  >>>   LAUNCH TIME
      COMMON LTIME,LTIN,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME
C
C  >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C  >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C  >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK.RNGSMC,RNGSMD
C
C  >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATH,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(SPMAX,VPAR(18))
      INTEGER ODCR(144),OBUF(557)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DIMENSION NAME(3)
      EQUIVALENCE (OBUF(1),PI)
      CALL OPEN(ODCB,IERR,NAME,0)
      IF(IERR .LT. 0)CALL CREAT(ODCB,IERR,NAME,14,3,0,0,144)
      IF(JJ.EQ.1)CALL WRITF(ODCB,IERR,OBUF,557)
      IF(JJ.EQ.0)CALL READF(ODCB,IERR,OBUF,557)
      CALL CLOSE(ODCB,IERR)
      RETURN
      END
      SUBROUTINE CPLOT(XDISTV,DOSV,CONCV,NUMV,IEXPD,IEXPC)
C
C   -------------------------------------------------------------------
C   -                                                                 -
C   -     THIS SUBROUTINE PLOTS THE DOSAGE AND CONCENTRATION CENTERLINE -
C   -     CURVES                                                      -
C   -                                                                 -
C   -------------------------------------------------------------------
C
C
C   >>>   COMMON AREAS
C
C
C   >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C   >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHEL
C
C   >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNDFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,MONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C   >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C   >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C   >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C   >>>   LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME
C
C   >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIN,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C   >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C   >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD

    >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
```

```
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PHO,VPAR(17)),(GAMMAX,VPAR(18))
C
C        DIMENSION STATEMENT
C
      DIMENSION XDISTV(1),DOSV(1),CONCV(1)
C
C        CALCULATE PLOTTING FACTORS
C
      FXDIST = 9295.0/30000.0
      FDOS = 8231.0 * 10.0**(IEXPD - 1)
      FCONC = 8231.0 * 10.0**(IEXPC - 1)
C
C        PLOT THE DOSAGE CENTERLINE CURVE
C
      DO 7 I=1,NUMV
      J = 1/I
      J = 1 - 2 * J
      IXPLOT = XDISTV(I) * FXDIST + 775.0
      IYPLOT = DOSV(I) * FDOS + 1040.0
      WRITE (IPL2) J,1,IXPLOT,IYPLOT
    7 CALL SYMBL(100,100,25400B,IPL2)
C
C        PLOT THE CONCENTRATION CENTERLINE CURVE
C
      DO 16 I=1,NUMV
      J = 1/I
      J = 1 - 2 * J
      IXPLOT = XDISTV(I) * FXDIST + 775.0
      IYPLOT = CONCV(I) * FCONC + 1046.0
   16 WRITE (IPL2) J,1,IXPLOT,IYPLOT
C
C        RETURN TO RCONC
C
      RETURN
C
C        END OF CPLOT
C
      END
      SUBROUTINE LABEL(IEXPD,IEXPC)
C
C    -----------------------------------------------------------------
C    -                                                               -
C    -   THIS SUBROUTINE LABELS THE CONCENTRATION AND DOSAGE CENTERLINE -
C    -   PLOTS                                                       -
C    -                                                               -
C    -----------------------------------------------------------------
C
C
C  >>>   COMMON AREAS
C
C
C  >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQRPI,DEGRAD,RADDEG,IV2
C
C  >>>   VEHICLE PARAMETERS
```

```
      COMMON VPAR(18),SIGHCL
C
C >>>    OPTION PARAMETERS
      COMMON YPOS,YES,ISNDFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTHP,
     .        NUMRUN,LU,NUM,NONLY,MFORM,LUPRNT,ITGU,IPL1,IPL2,IPL3
      INTEGER YES
C
C >>>    MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .        SURDEN,FILE(3)
      INTEGER FILE
C
C >>>    CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C >>>    SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C >>>    LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME
C
C >>>    CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .        TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C >>>    DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .        CLDLNG,AVSPD,AVDIR,SIGAZ
C
C >>>    CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C >>>    EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
C
C        OUTPUT FORMAT STATEMENTS
C
  200 FORMAT (4I5,I2)
  201 FORMAT (4I5,F5.0)
  202 FORMAT (4I5,F5.2)
  203 FORMAT (4I5,I4,1XA1,A2,1XI2,1XA2,A1,1XI4)
  204 FORMAT (4I5,I4" C"A2,2XI2,1XA2,A1,1XI4)
  205 FORMAT (4I5,I4,1XR1,A2,2XI2,1XA2,A1,1XI4)
C
C        LABEL THE PLOT
C
      I = - IEXPC
      WRITE (IPL2) -1,1,300,4800
      WRITE (IPL2,200) 0,150,-100,0,I
      I = - IEXPD
      WRITE (IPL2) -1,1,300,6300
      WRITE (IPL2,200) 0,150,-100,0,I
      WRITE (IPL2) -1,1,3700,[illegible]
      [illegible]
```

```
      WRITE (IPL2) -1,1,3700,8745
      WRITE (IPL2,201) 125,0,0,125,STBTIM
      WRITE (IPL2) -1,1,3700,8540
      WRITE (IPL2,202) 125,0,0,125,CONMAX
      WRITE (IPL2) -1,1,3700,8335
      WRITE (IPL2,201) 125,0,0,125,TOPLAY
      WRITE (IPL2) -1,1,3700,8130
      WRITE (IPL2,201) 125,0,0,125,BOTLAY
      WRITE (IPL2) -1,1,3700,7925
      WRITE (IPL2,201) 125,0,0,125,CALHT
      IF(ISNDFO .EQ. 1)GO TO 4
      WRITE (IPL2) -1,1,5625,9000
      WRITE (IPL2) 1,1,6125,9000
      GO TO 7
    4 WRITE (IPL2) -1,1,5025,9000
      WRITE (IPL2) 1,1,5525,9000
    7 WRITE (IPL2) -1,1,6225,8950
      WRITE (IPL2,203) 125,0,0,125,ISTIME,ITIMEZ,LAUNTD(4),ISDAY,ISMON,
     .                ISYEAR
      WRITE (IPL2) -1,1,5725,8745
      WRITE (IPL2,204) 125,0,0,125,ITIME,LAUNTD(4),IDAY,IMONTH,IYEAR
      WRITE (IPL2) -1,1,5725,8540
      IF(LTIME)WRITE(IPL2,205)205,0,0,205,LTIM,LAUNTD(3),LAUNTD(4),LDAY,
     .                          LMONTH,LYEAR
C
C        RETURN TO RCONC
C
      RETURN
C
C        END OF LABEL
C
      END
      SUBROUTINE DFEXP(J,CONC)
C
C   ---------------------------------------------------------------------
C   -                                                                   -
C   -    THIS SUBROUTINE CALCULATES DIFFUSION EXPONENTIALS              -
C   -                                                                   -
C   -    J - INDEX IN THE ALT ARRAY OF THE TOP OF THE LAYER             -
C   -    CONC - CONCENTRATION TO BE TESTED                              -
C   -                                                                   -
C   ---------------------------------------------------------------------
C
C   >>>    COMMON AREAS
C
C
C   >>>    MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C   >>>    VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C   >>>    OPTION PARAMETERS
      COMMON YPOS,YES,ISNDFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,MONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C   >>>    MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),[illegible]
     .       SURDEN,FILE(3)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      INTEGER FILE
C
C >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C >>>   LAUNCH TIME
      COMMON LTIME,LTIN,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME
C
C >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIN,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       TCLNG,AVSPD,AVDIR,SIGAZ
C
C >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))

C
C       CALCULATE SIGMA Z
C
      SIGZ = XDIST * SIGAP + SIGZ0/1.28
      SQSIGZ = 2.0 * SIGZ * SIGZ
C
C       CALCULATE THE EXPONENTIAL SUM IN THE DIFFUSION EQUATION
C
      TWOI = 2.0
      ZT = ALT(J)
      TEMP2 = STBHT - CALHT
      TEMP3 = STBHT - 2.0 * BOTLAY + CALHT
      EXPZ = EXP( - TEMP2 * TEMP2/SQSIGZ) +
     .       EXP( - TEMP3 * TEMP3/SQSIGZ)
    4 TEMP1 = TWOI * (ZT - BOTLAY)
      TEXPSM = EXPZ
      TEXP = (TEMP1 - TEMP2)**2/SQSIGZ
      IF(TEXP .LE. 120.0)EXPZ = EXPZ + EXP( - TEXP)
      TEXP = (TEMP1 + TEMP2)**2/SQSIGZ
      IF(TEXP .LE. 120.0)EXPZ = EXPZ + EXP( - TEXP)
      TEXP = (TEMP1 - TEMP3)**2/SQSIGZ
      IF(TEXP .LE. 120.0)EXPZ = EXPZ + EXP( - TEXP)
      TEXP = (TEMP1 + TEMP3)**2/SQSIGZ
      IF(TEXP .LE. 120.0)EXPZ = EXPZ + EXP( - TEXP)
      IF(EXPZ .EQ. TEXPSM)GO TO 7
      TWOI = TWOI + 2.0
      GO TO 4
    7 EXPZ = REFLEC * EXPZ
```

```
C         CALCULATE SIGMA Y
C
CC
CC***THE DIFFERENCE IN THE WIND DIRECTIONS USED IN THE ALGORITHM
CC***FOR STANDARD DEVIATION OF THE LATERAL CONCENTRATION MUST BE
CC***LESS THAN 180 DEGREES...THE FOLLOWING TWO LINES OF CODE
CC***HAVE BEEN ADDED
      ITHET = IABS(IDIR(J) - IDIR(I))
      IF(ITHET.GE.180) ITHET = 360 - ITHET
      SIGAL = XDIST * SIGAP + SIGX0
      SIGY = SQRT(SIGAL * SIGAL +
     .            (0.0040589 * FLOAT(ITHET) * XDIST)**2)
C
C         CALCULATE CLOUD LENGTH
C
      DELVEL = SPEED(J) - SPEED(I)
      CLDLNG = 0.28 * DELVEL * XDIST/AVSPD
      IF(DELVEL .GE. 0.0)GO TO 11
      IF(PTEMP(J)-PTEMP(I) .GT. 0.0)CLDLNG = 0.0
      CLDLNG = ABS(CLDLNG)
C
C         CALCULATE SIGMA X
C
   11 SIGX = SQRT((CLDLNG/4.3)**2 + SIGX0 * SIGX0)
C
C*** CALCULATE TOTAL CLOUD LENGTH...THIS PARAMETER REPLACES CLDLNG IN COMMON
C*** CRAMER DOCUMENTAION IN ERROR...SEE STEPHENS/BJORK/HIPPS PERSONAL COPY
      TCLNG = 4.3 * SIGX
C
C         IF CONC=1000.0, DO NOT CALCULATE CROSS WIND DISTANCE BUT RETURN
C         TO THE CALLING PROGRAM
C
      IF(CONC .EQ. 1000.0)GO TO 20
C
C         CALCULATE CROSS WIND DISTANCE
C
      YDIST = - 2.0 * SIGY * SIGY * ALOG(15.7496 * CONC * SIGX * SIGY *
     .                           SIGZ/(SIGHCL * EXPZ))
      YDIST = SQRT(AMAX1(YDIST,0.0))
C
C         RETURN TO THE CALLING PROGRAM
C
   20 CONTINUE
C     WRITE(6,98)
   98 FORMAT(6X,"TL",1X,"DELVEL",6X,"CONC",7X,"AVSPD",7X,"SIGAL",
     .       7X,"SIGAP",7X,"SIGX0",2X,"J",3X,"PT(J)",3X,"PT(I)",
     .       3X,"TEMP2",3X,"TEMP3")
C     WRITE(6,99) TCLNG,DELVEL,CONC,AVSPD,SIGAL,SIGAP,SIGX0,J,PTEMP(J),
C    .            PTEMP(I),TEMP2,TEMP3
   99 FORMAT(1X,0PF8.1,F6.2,1P3E12.4,2E12.4,I3,0P4F8.2)
      RETURN
C
C         END OF DFEXP
C
      END
      SUBROUTINE ORGIN(IX0,IY0,LVERSN)
C
C ---------------------------------------------------------------------
C -
C -       THIS SUBROUTINE GIVES THE APPROPRIATE COORDINATES FOR PLOTTING
```

```
C  -      FOR THE COMPLEX AND MAP SELECTED                                    -
C  -                                                                          -
C  ----------------------------------------------------------------------------
C
C
C  >>>   COMMON AREAS
C
C
C  >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C  >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNDFO,IVHICL,IRUN,ICALC,IPLACE,ITIME2,IGOTMP,
     .       NUMRUN,LU,NUM,MONLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C  >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C  >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C  >>>   LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME
C
C  >>>   CLOUD . ABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPT2
C
C  >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C  >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C  >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATH,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      DIMENSION ILINE(32),IDATAF(10),IERS(32),IMAPL(48),NAMEF(3)
C
C        INPUT FORMAT STATEMENT
C
  100 FORMAT (I2,1XA1)
C
C        OUTPUT FORMAT STATEMENT
```

```
C
C        DIMENSION STATEMENT
C
      DIMENSION IX(8),IY(8)
C
C        DATA STATEMENTS
C
      DATA IERS/32*2H  /
      DATA NAMEF/2H?R,2HIS,2H02/
      DATA IMAPL/2H40,2H,S,2HEA,2H M,2HAP,2H  ,
     1           2H40,2H,L,2HAN,2HD ,2HMA,2HP ,
     1           2H41,2H,S,2HEA,2H M,2HAP,2H  ,
     1           2H4?,2H,L,2HAN,2HD ,2HMA,2HP ,
     1           2H1?,2H,S,2HEA,2H M,2HAP,2H  ,
     1           2H17,2H,L,2HAN,2HD ,2HMA,2HP ,
     1           2H39,2H,S,2HEA,2H M,2HAP,2H  ,
     1           2H39,2H,L,2HAN,2HD ,2HMA,2HP /
      DATA LCHAR/1HL/
      DATA IX/5459,5411,4030,4025,8750,8730,4100,4100/
      DATA IY/2630,8243,2465,8050,2990,8600,1700,7300/
C
C        IS THIS THE FIRST TIME THROUGH THIS SUBROUTINE? --
C        IF NOT, IT IS NOT NECESSARY TO CALCULATE THE INDEX OF THE
C        COORDINATES, I, AGAIN
C
      IF(IGOTMP .NE. 0)GO TO 7
C
C        THIS IS THE FIRST TIME THROUGH -- READ IN THE COMPLEX NUMBER
C        AND THE DESIRED MAP, i.e. SEA OR LAND
C
      CALL DREAD(NAMEF,7,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 16.
      IF(LVERSN.EQ.0)GO TO 10
      IF(YPOS.LT.48.0) YPOS = 458.
   10 IF(IVHICL.EQ.1) CALL DREAD(NAMEF,8,ILINE)
      IF(IVHICL.EQ.2) CALL DREAD(NAMEF,9,ILINE)
      IF(IVHICL.EQ.0) CALL DREAD(NAMEF,10,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(24.,YPOS,0,ILINE(2),8,3,0)
      CALL CHAR(95.,YPOS,0,ILINE(7),50,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,31,0,31,IXC,IYC)
      CALL CHAR(0.,YPOS,0,IERS,64,0,0)
      CALL CHAR(200.,YPOS+16,0,IERS,25,0,0)
      IF(IXC.LT.6.AND.IVHICL.EQ.1) I=1
      IF(IXC.GT.5.AND.IXC.LT.12.AND.IVHICL.EQ.1) I=2
      IF(IXC.GT.11.AND.IXC.LT.18.AND.IVHICL.EQ.1) I=3
      IF(IXC.GT.17.AND.IVHICL.EQ.1) I=4
      IF(IXC.LT.6.AND.IVHICL.EQ.2) I=5
      IF(IXC.GE.6.AND.IVHICL.EQ.2) I=6
      IF(IXC.LT.6.AND.IVHICL.EQ.0) I=7
      IF(IXC.GE.6.AND.IVHICL.EQ.0) I=8
      IMP = (I - 1)*6 + 1
      CALL CHAR(200.,YPOS+16.,0,IMAPL(IMP),12,0,0)
      YPOS = YPOS - 16.
      IF(LVERSN.EQ.0)GO TO 6
      IF(YPOS .LT.48.0) YPOS = 45?
C
C        SET THE COORDINATES BASED ON THE INDEX I
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
    6 IGOTMP = I
    7 CONTINUE
      I = IGOTMP
      IX0 = IX(I)
      IY0 = IY(I)
C
C        RETURN TO THE CALLING PROGRAM
C
      RETURN
C
C        END OF ORGIN
C
      END
      SUBROUTINE SYMBL(IWIDE,IHI,ISYMB,LUPLT)
C
C   -------------------------------------------------------------------
C   -                                                                 -
C   -  THIS SUBROUTINE PLOTS A GIVEN SYMBOL A SPECIFIED WIDTH AND     -
C   -  HEIGHT.                                                        -
C   -                                                                 -
C   -------------------------------------------------------------------
C
      IX=-IWIDE/2
      IY=-IHI/2
      WRITE(LUPLT) -1,-1,IX,IY
      WRITE(LUPLT,100) IWIDE,0,0,IHI,ISYMB
  100 FORMAT(4I5,A1,1H_)
      IY=-IY
      WRITE(LUPLT)-1,-1,IX,IY
      RETURN
      END
      SUBROUTINE DREAD(NAMEF,LNUM,ILINE)
C
C   -------------------------------------------------------------------
C   -                                                                 -
C   -  THIS SUBROUTINE READS A SPECIFIED QUESTION FROM A SPECIFIED    -
C   -  QUESTION FILE. THE QUESTION IS THEN DISPLAYED UPON THE         -
C   -  PLASMASCOPE FOR PROCESSING.                                    -
C   -                                                                 -
C   -------------------------------------------------------------------
C
      DIMENSION NAMEF(3),IDCB(144),IBUF(40),ILINE(32)
      CALL OPEN(IDCB,IERR,NAMEF,0)
      LOOP = LNUM - 1
      DO 10 I=1,LOOP
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
   10 CONTINUE
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,100) (ILINE(I),I=1,32)
  100 FORMAT(32A2)
      CALL CLOSE(IDCB,IERR)
      RETURN
      END
      SUBROUTINE BLANK(IBUF,II)
C
C   -------------------------------------------------------------------
```

```
C  -                                                                  -
C  -   THIS SUBROUTINE FILLS A SPECIFIED ARRAY WITH A SPECIFIED      -
C  -   NUMBER OF BLANKS.                                              -
C  -                                                                  -
C  -----------------------------------------------------------------
C
       DIMENSION IBUF(40)
       DATA IBLK/2H  /
       DO 10 I=1,II
 10    IBUF(I) = IBLK
       RETURN
       END
       SUBROUTINE LERS(YPOS,LVERSN)
C
C  -----------------------------------------------------------------
C  -                                                                  -
C  -   THIS SUBROUTINE ERASES A SPECIFIED LINE OF 64-CHARACTERS     -
C  -   FROM THE PLASMASCOPE.                                          -
C  -                                                                  -
C  -----------------------------------------------------------------
C
       DIMENSION IERS(32)
       DATA IERS/32*2H  /
       IF(YPOS .GT. 48.0)GO TO 4
       IF(LVERSN.EQ.0)GO TO 4
       YPOS = 458.0
       CALL CLEAR
       RETURN
 4     CALL CHAR(0.0,YPOS,0,IERS,64,0,0)
       CALL CHAR(0.0,YPOS-16.0,0,IERS,64,0,0)
       RETURN
       END
```

&RISOS T=00003 IS ON CR00003 USING 00068 BLKS R=0000

```
FTN4.L
      PROGRAM RISOS(3),REEDS ISOPLETH PLOTTING PROGRAM -- 2/3/79
C
C     ******************************************************************
C     *                                                                *
C     *     ISOPLETH PLOTTING PROGRAM -- A PROGRAM IN THE REEDS SERIES *
C     *     OF PROGRAMS                                                *
C     *                                                                *
C     ******************************************************************
C
C
C  >>>   COMMON AREAS
C
C
C  >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C  >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNOFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .        NUMRUN,LU,NUM,NOMLY,MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .        SURDEN,FILE(3)
      INTEGER FILE
C
C  >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C  >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C  >>>   LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME, LVERSN
C
C  >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .        TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C  >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .        CLDLNG,AVSPD,AVDIR,SIGAZ
C
C  >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C  >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATH,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            ([illegible] VPAR([illegible])),([illegible] VPAR([illegible])),([illegible],VPAR([illegible])),
     .            (P[illegible]3,VPAR(16)),([illegible]0,VPAR(17)),(GAMMA[illegible] [illegible]R(18))
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C         OUTPUT FORMAT STATEMENTS
C
199   FORMAT(A1)
  200 FORMAT ("1"20X"CLOUD LOCATION AND DIMENSIONS"/
     .        7X"TIME FROM LAUNCH"12X"RANGE"5X"AZIMUTH"
     .        8X"DIAMETERS (METERS)"/
     .        6X"(MIN)"9X"(SEC)"9X"(METERS)"4X"(DEG)"6X"CROSS WIND"
     .        4X"ALONG WIND")
  201 FORMAT (7XI3,10XF5.1,9XF8.1,4XF5.1,7XF7.1,7XF7.1)
  202 FORMAT (4I5,I4" C"A2,2XI2,1XA2,A1,1XI4)
  203 FORMAT (4I5,A1)
  204 FORMAT (4I5,F5.2"_")
  205 FORMAT (4I5", "F5.2"_")
  206 FORMAT (F5.3)
  207 FORMAT (I1)
  208 FORMAT (4I5,I4,1XR1,A2,2XI2,1XA2,A1,1XI4)
C
C         TYPE AND DIMENSION STATEMENTS
C
      LOGICAL DFALTC
      DIMENSION CONC(10),NAME(3),NAMEF(3),ILINE(32),IDATAF(10),
     .           IERS(32),IXA(100),IYA(100),IXB(100),IYB(100)
      DATA NAME/2H=R,2HEE,2HD2/, NAMEF/2H?R,2HIS,2HOS/
      DATA IERS/32*2H  /
C
C         CALL GRAF TO INITIALIZE SCOPE (APPROPRIATE ONLY WHEN USING
C         PLASMASCOPE)
C
      CALL GRAF(1)
C
C         FIND OUT IF THIS IS THE CRT OR PLASMASCOPE VERSION OF REED2
C         BEING RUN
C
      CALL VERSN(I)
      LVERSN = I .EQ. 0
C
C         READ COMMON DISK FILE
C
      CALL RWDIS(NAME,0)
C
C         DETERMINE THE ORIGIN ON THE MAP FOR THIS PLOT AND MOVE THE
C         PEN THERE
C
      CALL ORGIN(IX0,IY0,LVERSN)

      WRITE (IPL3) -1,1,IX0,IY0
C
C         DETERMINE THE INDEX IN THE ALTITUDE DATA ARRAY THAT HAS
C         THAT ALTITUDE JUST LOWER THAN THE EFFECTIVE CLOUD HEIGHT, STBHT
C
      DO 4 I=2,LAYTOP
      IF(STBHT .GT. ALT(I))GO TO 4
      ISTBHT = I - 1
      GO TO 5
    4 CONTINUE
      ISTBHT = LAYTOP
C
C         DRAW AND LINE DEPICTING CLOUD MOVEMENT ALONG THE GROUND
C         AS FAR AS THE CLOUD STABILIZATION POINT
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
    5 X = 0.0
      Y = 0.0
      DO 9 I=2,ISTBHT
      IM1 = I - 1
      RANGE = 0.5 * (RISTIM(I) - RISTIM(IM1)) * (SPEED(I) + SPEED(IM1))
C     DIR = 0.5 * FLOAT(IDIR(I) + IDIR(IM1))
      DIR = 0.5 * FLOAT(IDIR(I) + IDIR(IM1)) - 180.* (IABS( IDIR(I)
     .       - IDIR(IM1))/180)
C     IF(DIR .LE. 0) DIR = DIR + 360
      DIR = DEGRAD*DIR
      IFLIP = 1
      IF(DIR .GT. PI) IFLIP = -1
      DIR = DIR + IFLIP*PI
      X = X + RANGE * COS(PI - DIR)
      Y = Y + RANGE * SIN(PI - DIR)
      R = SQRT(X*X + Y*Y)
C     IF(IABS(IDIR(I) - IDIR(IM1)) .GT. 180)DIR = DIR - 180.0
C     DIR = 180. - DIR
C     IF(DIR .LT. 0.0)DIR = DIR + 360.0
C     DIR = DEGRAD * (360.0 - DIR)
C     DIR = DEGRAD * DIR
C     X = X + RANGE * COS(DIR)
C     Y = Y + RANGE * SIN(DIR)
      IX = INT(0.2631 * X) + IX0
      IY = INT(0.3545 * Y) + IY0
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 11
    9 WRITE (IPL3) 1,1,IX,IY
C
C        MAKE THE CALCULATIONS NECESSARY TO WRITE OUT THE CLOUD
C        LOCATION AND DIMENSIONS
C
   11 ALT1 = 0.5 * (STBHT + ALT(ISTBHT))
      ISTBP1 = ISTBHT + 1
      ARG1 = ALT(ISTBP1) - ALT(ISTBHT)
      ARG2 = (STBHT - ALT(ISTBHT))/ARG1
      SPCENT = SPEED(ISTBHT) + (SPEED(ISTBP1) - SPEED(ISTBHT)) * ARG2
CC
CC***SINCE THE EFFECTIVE CLOUD HEIGHT IS ONLY AN ARTIFICE BASED ON
CC***THE STABILIZATION HEIGHT, THE TIME TAKEN TO REACH THE EFFECTIVE
CC***CLOUD HEIGHT IS THE TOTAL RISE TIME, NOT A FRACTION OF THE
CC***RISE TIME...THE CODE HAS BEEN CHANGED ACCORDINGLY AFTER CONSUL-
CC***TATION WITH DR. STEPHENS...JSN
CC
      RANGE = SPCENT * (RISTIM(ISTBP1) - RISTIM(ISTBHT))
      IF(IABS(IDIR(ISTBP1) - IDIR(ISTBHT)) .LT. 180)GO TO 14
      IF(IDIR(ISTBP1) .LT. 180)IDIR(ISTBP1) = IDIR(ISTBP1) + 360
      IF(IDIR(ISTBHT) .LT. 180)IDIR(ISTBHT) = IDIR(ISTBHT) + 360
CCC14 DIR = FLOAT(IDIR(ISTBHT)) + (ALT1 - ALT(ISTBHT)) *
CCC  .                          FLOAT(IDIR(ISTBP1) - IDIR(ISTBHT))/ARG1
   14 DIR = FLOAT(IDIR(ISTBHT)) + ((STBHT - ALT(ISTBHT))/ARG1)
     .       * FLOAT(IDIR(ISTBP1) - IDIR(ISTBHT))
CCC   IF(DIR .GT. 360.0)DIR = DIR - 360.0
CCC   IF(DIR .GT. 180.0)GO TO 17
CCC   DIR = DIR + 180.0
CCC   GO TO 18
CCC17 DIR = DIR - 180.0
CCC18 DIR = 180.0 - DIR
      IF(DIR .GT 360.) DIR = DIR - 360.0
      ..P = ...
```

```
      IF(DIR.GT.180.0) FLIP = -1.0
      DIR = DIR + FLIP * 180.0
      ARG1 = DEGRAD * DIR
      X = X + RANGE * COS(PI - ARG1)
      Y = Y + RANGE * SIN(PI - ARG1)
      R = SQRT(X * X + Y * Y)
      DELR = 300.0 *AVSPD
      DACRS = 4.30 * SIGX0
      DALNG = 4.30 * SIGX0
C
      ANGL = ATAN2(Y,X)
      ANGL = ANGL * RADDEG
      ARG1 = 180.0
      IF(ANGL .GT. 180.0)ARG1 = 540.0
      AZ = ARG1 - ANGL
C
CCC   ARG1 = 180.0
CCC   IF(AVDIR .GT. 180.0)ARG1 = 540.0
CCC   DAZ = ARG1 - AVDIR
CCC   ARG1 = DEGRAD * DAZ
      IF(AVDIR.GE.90.0) DAZ = 180.0 - AVDIR
      IF(AVDIR.GE.180.0) DAZ = AVDIR - 180.0
      IF(AVDIR.GE.270.0) DAZ = 360.0 - AVDIR
      ARG1 = DEGRAD * DAZ
      DELX = DELR * COS(ARG1)
      DELY = DELR * SIN(ARG1)
C
C  INITIALIZE SOME QUANTITIES FOR THE SUBSEQUENT DO LOOP
C
      IDELTH = IABS(IDIR(LAYTOP) - IDIR(1))
      IF(IDELTH.GE.180)IDELTH = 360 - IDELTH
C
      DELU = ABS(SPEED(ISTBHT) - SPEED(1))
C
      TIM = STBTIM
      ITIM = INT(TIM)/60
      RE = AMOD(TIM,60.0)
      XDIST = 0.0
      XC = X
      YC = Y
      TXL = 0.28 * DELU/AVSPD
      SIGX02 = SIGX0 * SIGX0
      SIGAL2 = SIGAL * SIGAL
      WRITE (LUPRNT,200)
CC
CC*** IN THE FOLLOWING DO LOOP, "STBRNG IS THE DISTANCE FROM THE
CC*** PAD TO STABILIZATION, AND "R" IS THE DISTANCE FROM THE PAD
CC*** XDIST IS THE DISTANCE FROM THE STABILIZATION POINT.
CC
      DO 22 I=1,13
      WRITE (6,666) XDIST
  666 FORMAT(" THE DISTANCE FROM STABILIZATION IS:",F12.2)
      WRITE (LUPRNT,201) ITIM,RE,R,AZ,DACRS,DALNG
      XDIST = XDIST + DELR
C     TIM = TIM + 5.0
      TIM = TIM + 300.0
      ITIM = INT(TIM)/60
      RE = AMOD(TIM,60.0)
      XL = XDIST + TXL
      SIGX = SQRT((XL/4.30)**2 + SIGX02)
```

```
      DALNG = 4.30 * SIGX
CC
CC***THE ALGORITHM FOR SIGY HAS BEEN CHANGED TO CONFORM TO THE
CC***ALGORITHM FOR SIGY FOUND IN SUBROUTINE DFEXP AFTER CONSUL-
CC***TATION WITH DR. STEPHENS...JSH
CC
      SIGAL2 = (SIGAP*XDIST + SIGX0)**2
      SIGY = SQRT(SIGAL2 + (0.0040589 * FLOAT(IDELTH) * XDIST)**2)
      DACRS = 4.36 * SIGY
      XC = XC + DELX
      YC = YC + DELY
      R = SQRT(XC * XC + YC * YC)
   22 AZ = 180.0 - RADDEG * ATAN2(YC,XC)
C
C        LABEL THE CLOUD STABILIZATION POINT WITH A +
C
      IX = INT(0.2631 * X) + IX0
      IY = INT(0.3545 * Y) + IY0
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 77
      IXX = IX
      IYY = IY
      WRITE (IPL3) 1,1,IX,IY
      CALL SYMBL(150,150,1H+,IPL3)
C
C        LABEL THE POINT OF MAXIMUM CONCENTRATION WITH A 0
C
      DIR = DEGRAD * (180.0 - AVDIR)
      CDIR = COS(DIR)
      SDIR = SIN(DIR)
      IX1 = INT(0.2631 * (X + RNGSMC * CDIR)) + IX0
      IY1 = INT(0.3545 * (Y + RNGSMC * SDIR)) + IY0
      WRITE (IPL3) -1,1,IX1,IY1
      CALL SYMBL(150,150,1H0,IPL3)
C
C        DRAW THE LINE OF CLOUD MOVEMENT ALONG THE GROUND FROM
C        THE CLOUD STABILIZATION POINT ON
C
      WRITE (IPL3) -1,1,IXX,IYY
      RANGE = 1000.0
   27 X = X + RANGE * CDIR
      Y = Y + RANGE * SDIR
      IX = INT(0.2631 * X) + IX0
      IY = INT(0.3545 * Y) + IY0
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 29
      WRITE (IPL3) 1,1,IX,IY
      GO TO 27
   29 WRITE (IPL3) -1,1,IXX,IYY
C
C        ARE DEFAULT CONCENTRATION VALUES GOING TO BE USED
C        FOR THE PLOTS
C
CC    IF(YPOS.LT.48.0) YPOS = 458.
      IF(YPOS.LT.48.0 .AND. LVERSN) YPOS = 458.
      CALL DREAD(NAMEF,2,ILINE)
CC    CALL LERS(YPOS,LVERSN)
      IF(LVERSN) CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 16.
      CALL DREAD(NAMEF,3,ILINE)
         LEI    YOS,.....SN).
```

```
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 32.
C
C        YES -- SET UP THE DEFAULT VALUES
C
      CONC(1) = 0.1 * CONMAX
      CONC(2) = 0.5 * CONMAX
      CONC(3) = 0.75 * CONMAX
      CONC(4) = - 1.0
      CALL CODE
      WRITE (IDATAF,206) CONC(1)
      CALL CHAR(440.,YPOS+48.,0,IDATAF,5,0,0)
      CALL CODE
      WRITE (IDATAF,206) CONC(2)
      CALL CHAR(120.,YPOS+32.,0,IDATAF,5,0,0)
      CALL CODE
      WRITE (IDATAF,206) CONC(3)
      CALL CHAR(256.,YPOS+32.,0,IDATAF,3,0,0)
      CALL DREAD(NAMEF,4,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.,YPOS,0,ILINE,46,3,0)
      CALL CHAR(384.,YPOS,0,ILINE(25),8,3,0)
      CALL CHAR(464.,YPOS,0,ILINE(30),6,0,0)
      IF(LVERSN) GO TO 33
      READ(LU,199)IANS
      IX = 25
      IF(IANS.NE.1HN) GO TO 34
      IX = 30
      GO TO 34
   33 CONTINUE
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IX,IY)
   34 CONTINUE
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      IF(IX .LE. 25)CALL CHAR(464.,YPOS,0,IERS,6,0,0)
      IF(IX .GT. 25)CALL CHAR(384.,YPOS,0,IERS,8,0,0)
      YPOS = YPOS - 32.
CC    IF(YPOS .LT. 64.0)YPOS = 458.0
      IF(YPOS .LT. 64.0 .AND. LVERSN)YPOS = 458.0
      DFALTC = .FALSE.
      IF(IX .LT. 20)DFALTC = .TRUE.
C
C        DO LOOP OVER THE 10 POSSIBLE CONCENTRATION VALUES FOR THE PLOTS
C
      IF(DFALTC)GO TO 35
      CALL DREAD(NAMEF,5,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 32.
      IF(YPOS.LE.64) YPOS=458.
   35 DO 53 I=1,10
C
C        IF DEFAULT CONCENTRATION VALUES ARE NOT BEING USED,
C        READ IN THE VALUE FOR THIS PLOT
C
      IF(DFALTC)GO TO 37
      CALL DREAD(NAMEF,6,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.,YPOS,0,ILINE,'7,3,0)
      CALL CODE
      WRITE (IDX,207) I
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CALL CHAR(111.,YPOS,0,IDX,1,3,0)
      NIN = 9
      CALL BLANK(IDATAF,10)
      CALL IN(0,JTYPE,144.,YPOS,0,IDATAF,NIN,0,31,0,31,IX,IY)
      CALL CODE
      READ (IDATAF,*) CONC(I)
      CALL CHAR(0.,YPOS,0,ILINE,17,0,0)
      CALL CHAR(111.,YPOS,0,IDX,1,0,0)
      YPOS = YPOS - 16..
CC    IF(YPOS .LT. 40.0)YPOS = 458.0
      IF(YPOS .LT. 40.0 .AND. LVERSN)YPOS = 458.0
   37 IF(CONC(I) .LT. 0.0)GO TO 61
C
C        ITERATE TO FIND THE LOCATION OF THIS CONCENTRATION
C        ON THE PLOT
C
      XDIST = 0.0
      DINC = 1000.0
C
   41 CALL DFEXP(LAYTOP,CONC(I))
      IF(YDIST .GT. 0.0)GO TO 42
      XDIST = XDIST + DINC
      GO TO 41
C
   42 IF(DINC .LE. 100.0)GO TO 43
      XDIST = XDIST - 900.0
      DINC = 100.0
      GO TO 41
C
   43 IF(DINC .LE. 10.0)GO TO 44
      XDIST = XDIST - 90.0
      DINC = 10.0
      GO TO 41
C
C        PLOT OUT THE CONCENTRATION LINE ON BOTH SIDES
C
   44 XDIST = XDIST - 10.0
      IX = INT(0.2631 * XDIST * CDIR) + IXX
      IY = INT(0.3545 * XDIST * SDIR) + IYY
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 59
      NUMA = 1
      IXA(NUMA) = IX
      IYA(NUMA) = IY
      NUMB = 1
      IXB(NUMB) = IX
      IYB(NUMB) = IY
C
      XDIST = XDIST + 10.0
      IX = INT(0.2631 * (XDIST * CDIR - YDIST * SDIR)) + IXX
      IY = INT(0.3545 * (XDIST * SDIR + YDIST * CDIR)) + IYY
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 59
      NUMA = 2
      IXA(NUMA) = IX
      IYA(NUMA) = IY
C
      IX = INT(0.2631 * (XDIST * CDIR + YDIST * SDIR)) + IXX
      IY = INT(0.3545 * (XDIST * SDIR - YDIST * CDIR)) + IYY
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 54
      NUMB = 2
         NUM     IX
```

```
      IYB(NUMB) = IY
C
   46 XDIST = XDIST + 500.0
      CALL DFEXP(LAYTOP,CONC(I))
      IX = INT(0.2631 * (XDIST * CDIR - YDIST * SDIR)) + IXX
      IY = INT(0.3545 * (XDIST * SDIR + YDIST * CDIR)) + IYY
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 54
      NUMA = NUMA + 1
      IXA(NUMA) = IX
      IYA(NUMA) = IY
C
      IF(YDIST .GT. 0.0)GO TO 52
      NUMB = NUMB + 1
      IXB(NUMB) = IX
      IYB(NUMB) = IY
      GO TO 54
C
   52 IX = INT(0.2631 * (XDIST * CDIR + YDIST * SDIR)) + IXX
      IY = INT(0.3545 * (XDIST * SDIR - YDIST * CDIR)) + IYY
      IF(IX.LT.0 .OR. IX.GT.9999 .OR. IY.LT.0 .OR. IY.GT.9999)GO TO 54
      NUMB = NUMB + 1
      IXB(NUMB) = IX
      IYB(NUMB) = IY
      GO TO 46
C
   54 WRITE (IPL3) -1,1,IXA(1),IYA(1)
      DO 56 J=2,NUMA
   56 WRITE (IPL3) 1,1,IXA(J),IYA(J)
      IF(NUMB .EQ. 1)GO TO 59
      WRITE (IPL3) -1,1,IXB(1),IYB(1)
      DO 57 J=2,NUMB
   57 WRITE (IPL3) 1,1,IXB(J),IYB(J)
C
   59 CONTINUE
C
C        ON THE PLOT, CROSS OUT EITHER THE WORD FORECAST OR SOUNDING
C
   61 IF(ISNDFO .NE. 0)GO TO 62
      WRITE (IPL3) -1,1,707,604
      WRITE (IPL3) 1,1,1174,604
      GO TO 64
C
   62 WRITE (IPL3) -1,.,1269,604
      WRITE (IPL3) 1,1,1760,604
C
C        PRINT OUT THE PREDICTION TIME ON THE PLOT
C
   64 WRITE (IPL3) -1,1,1069,319
      WRITE (IPL3,202) 100,0,0,150,ITIME,LAUNTD(4),IDAY,IMONTH,IYEAR
C
C        IF THE LAUNCH TIME WAS ENTERED, PRINT IT OUT ON THE PLOT
C
      IF(.NOT. LTIME)GO TO 67
      WRITE (IPL3) -1,1,1669,112
      WRITE (IPL3,208) 100,0,0,150,LTIM,LAUNTD(3),LAUNTD(4),LDAY,
     .                  LMONTH,LYEAR
C
C        ON THE PLOT, PRINT OUT THE CHARACTERS + AND O FOR THE LEGEND
C
   67 WRITE (IPL3) -1,1,1041,1342
```

```
      WRITE (IPL3,203) 150,0,0,150,1H+
      WRITE (IPL3) -1,1,1041,1104
      WRITE (IPL3,203) 150,0,0,150,1H0
C
C        FOR THE LEGEND ON THE PLOT, PRINT OUT THE CONCENTRATION VALUES
C        FOR WHICH CONTOURS WERE DRAWN
C
      WRITE (IPL3) -1,1,1066,9587
      DO 75 I=1,10
      IF(CONC(I) .LT. 0.0)GO TO 77
      IF(I .NE. 1)GO TO 72
      WRITE (IPL3,204) 125,0,0,150,CONC(I)
      GO TO 75
   72 WRITE (IPL3,205) 125,0,0,150,CONC(I)
   75 CONTINUE
C
C        WRITE OUT COMMON DISK FILE
C
   77 CALL RUDIS(NAME,1)
C
C        CALL NGRAF TO RETURN SCOPE TO NORMAL MODE OF OPERATION
C        (APPROPRIATE ONLY WHEN USING PLASMASCOPE)
C
      CALL NGRAF
C
C        RETURN TO THE MAIN PROGRAM REEDS
C
      STOP
C
C        END OF RISOS
C
      END
      END$
```

&RISOT T=00003 IS ON CR00003 USING 00062 BLKS R=0000

```
FTN4,L
      SUBROUTINE RWDIS(NAME,JJ)
C
C  -----------------------------------------------------------
C  -                                                          -
C  -   THIS SUBROUTINE READS AND WRITES COMMON AREAS TO DISC  -
C  -   TO BE UTILIZED BY THE VARIOUS REED PROGRAMS.           -
C  -                                                          -
C  -----------------------------------------------------------
C
C  >>>   COMMON AREAS
C
C
C  >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>   VEHICLE PARAMETERS
      COMMON VPAR(18),SIGHCL
C
C  >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNDFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,NOHLY,NFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C  >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C  >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C  >>>   LAUNCH TIME
      COMMON LTIME,LTIM,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME
C
C  >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C  >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SASIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C  >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C  >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL203,VPAR(16)),(PHO,VPAR(17)),[illegible]
      INTEGER ODCR(144),OBUF(557)
```

ORIGINAL PAGE IS OF POOR QUALITY

```
      DIMENSION NAME(3)
      EQUIVALENCE (OBUF(1),PI)
      CALL OPEN(ODCB,IERR,NAME,0)
      IF(IERR .LT. 0)CALL CREAT(ODCB,IERR,NAME,14,3,0,0,144)
      IF(JJ.EQ.1)CALL WRITF(ODCB,IERR,OBUF,557)
      IF(JJ.EQ.0)CALL READF(ODCB,IERR,OBUF,557)
      CALL CLOSE(ODCB,IERR)
      RETURN
      END
      SUBROUTINE DFEXP(J,CONC)
C
C  ------------------------------------------------------------------------
C  -                                                                      -
C  -     THIS SUBROUTINE CALCULATES DIFFUSION EXPONENTIALS                -
C  -                                                                      -
C  -     J - INDEX IN THE ALT ARRAY OF THE TOP OF THE LAYER               -
C  -     CONC - CONCENTRATION TO BE TESTED                                -
C  -                                                                      -
C  ------------------------------------------------------------------------
C
C  >>>   COMMON AREAS
C
C
C  >>>   MATH PARAMETERS
      COMMON PI,PIOVR2,PI43,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>   VEHICLE PARAMETERS
      COMMON VPAR(10),SIGHCL
C
C  >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNDFO,IVHICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .       NUMRUN,LU,NUM,NONLY,NFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .       SURDEN,FILE(3)
      INTEGER FILE
C
C  >>>   CALCULATION TIME
      COMMON ITIME,IDAY,INONTH(2),IYEAR
C
C  >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
C  >>>   LAUNCH TIME
      COMMON LTIME,LTIN,LDAY,LMONTH(2),LYEAR,LAUNTD(10)
      LOGICAL LTIME
C
C  >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C  >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       TCLNG,AVSPD,AVDIR,SIGAZ
C
C  >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMX,RNGSPK
C
```

```
C  >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATH,VPAR(10)),(HEATN,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))

C
C        CALCULATE SIGMA Z
C
      SIGZ = XDIST * SIGAP + SIGZ0/1.28
      SQSIGZ = 2.0 * SIGZ * SIGZ
C
C        CALCULATE THE EXPONENTIAL SUM IN THE DIFFUSION EQUATION
C
      TWOI = 2.0
      ZT = ALT(J)
      TEMP2 = STBHT - CALHT
      TEMP3 = STBHT - 2.0 * BOTLAY + CALHT
      EXPZ = EXP( - TEMP2 * TEMP2/SQSIGZ) +
     .       EXP( - TEMP3 * TEMP3/SQSIGZ)
    4 TEMP1 = TWOI * (ZT - BOTLAY)
      TEXPSM = EXPZ
      TEXP = (TEMP1 - TEMP2)**2/SQSIGZ
      IF(TEXP .LE. 120.0)EXPZ = EXPZ + EXP( - TEXP)
      TEXP = (TEMP1 + TEMP2)**2/SQSIGZ
      IF(TEXP .LE. 120.0)EXPZ = EXPZ + EXP( - TEXP)
      TEXP = (TEMP1 - TEMP3)**2/SQSIGZ
      IF(TEXP .LE. 120.0)EXPZ = EXPZ + EXP( - TEXP)
      TEXP = (TEMP1 + TEMP3)**2/SQSIGZ
      IF(TEXP .LE. 120.0)EXPZ = EXPZ + EXP( - TEXP)
      IF(EXPZ .EQ. TEXPSM)GO TO 7
      TWOI = TWOI + 2.0
      GO TO 4
    7 EXPZ = REFLEC * EXPZ
C
C        CALCULATE SIGMA Y
C
      SIGAL = XDIST * SIGAP + SIGX0
CC
CC***THE FOLLOWING TWO LINES WERE ADDED TO ENSURE THAT THE DIFF-
CC***ERENCE IN THE DIRECTIONS IS ALWAYS LESS THAN 180 DEGREES.
CC
      ITHET = IABS(IDIR(J) - IDIR(1))
      IF(ITHET.GT.180)ITHET = 360 - ITHET
      SIGY = SQRT(SIGAL * SIGAL +
     .            (0.0040589 * FLOAT(ITHET) * XDIST)**2)
C
C        CALCULATE CLOUD LENGTH
C
      DELVEL = SPEED(J) - SPEED(1)
      CLDLNG = 0.28 * DELVEL * XDIST/AVSPD
      IF(DELVEL .GE. 0.0)GO TO 11
      IF(PTEMP(J)-PTEMP(1) .GT. 0.0)CLDLNG = 0.0
      CLDLNG = ABS(CLDLNG)
C
C        CALCULATE SIGMA X
C
   11 SIGX = SQRT((CLDLNG/4.3)**2 + SIGX0 * SIGX0)
```

```
C
C*** CALCULATE TOTAL CLOUD LENGTH...THIS PARAMETER REPLACES CLDLNG IN COMMON
C*** CRAMER DOCUMENTAION IN ERROR...SEE STEPHENS/BJORK/HIPPS PERSONAL COPY
      TCLNG = 4.3 * SIGX
C
C        IF CONC=1000.0, DO NOT CALCULATE CROSS WIND DISTANCE BUT RETURN
C        TO THE CALLING PROGRAM
C
      IF(CONC .EQ. 1000.0)GO TO 20
C
C        CALCULATE CROSS WIND DISTANCE
C
      YDIST = - 2.0 * SIGY * SIGY * ALOG(15.7496 * CONC * SIGX * SIGY *
     .                         SIGZ/(SIGHCL * EXPZ))
      YDIST = SQRT(AMAX1(YDIST,0.0))
C
C        RETURN TO THE CALLING PROGRAM
C
20    CONTINUE
      RETURN
C
C        END OF DFEXP
C
      END
      SUBROUTINE ORGIN(IX0,IY0,LVERSN)
C
C  ----------------------------------------------------------------------
C  -                                                                    -
C  -   THIS SUBROUTINE GIVES THE APPROPRIATE COORDINATES FOR PLOTTING   -
C  -      FOR THE COMPLEX AND MAP SELECTED                              -
C  -                                                                    -
C  ----------------------------------------------------------------------
C
C
C   >>   COMMON AREAS

C
C  >>>   MATH PAR[illegible]
      COMMON PI,[illegible]143,TWOPI,SQR2PI,DEGRAD,RADDEG,IV2
C
C  >>>   VEHICLE [illegible]TERS
      COMMON VPAR[illegible]
C
C  >>>   OPTION PARAMETERS
      COMMON YPOS,YES,ISNDF[illegible]HICL,IRUN,ICALC,IPLACE,ITIMEZ,IGOTMP,
     .        N[illegible]RUN,LU,[illegible],MFORM,LUPRNT,ITDU,IPL1,IPL2,IPL3
      INTEGER YES
C
C  >>>   MET DATA
      COMMON ALT(30),IDIR(30),SPEED(30),TEMP(30),PRESS(30),PTEMP(30),
     .        SURDEN,FILE(3)
      INTEGER FILE
C
C  >>>   CALCULATION TIME
      COMMON ITIME,IDAY,IMONTH(2),IYEAR
C
C  >>>   SOUNDING TIME
      COMMON ISTIME,ISDAY,ISMON(2),ISYEAR
C
   >>    [illegible]
```

```
      COMMON LTIME,LTIN,LDAY,LMONTH(2),LYEAR,LAUNID(10)
      LOGICAL LTIME
C
C >>>   CLOUD STABILIZATION AND LAYER PARAMETERS
      COMMON STBALT,STBHT,STBAZD,STBRNG,STBTIM,CLDRAD,RISTIM(30),BOTLAY,
     .       TOPLAY,CALHT,LAYTOP,LAYBOT,REFLEC,IPTZ
C
C >>>   DIFFUSION COEFFICIENTS
      COMMON SIGX0,SIGX,SIGZ0,SIGAP,XDIST,YDIST,EXPZ,SQSIGZ,SIGY,SIGAL,
     .       CLDLNG,AVSPD,AVDIR,SIGAZ
C
C >>>   CONCENTRATION AND DOSAGE VALUES AND RANGES
      COMMON CONMAX,CONCPK,DOSMAX,DOSPK,RNGSMC,RNGSMD
C
C >>>   EQUIVALENCE STATEMENT FOR VEHICLE PARAMETERS
      EQUIVALENCE (QC1,VPAR(1)),(QC2,VPAR(2)),(QC3,VPAR(3)),
     .            (QT1,VPAR(4)),(QT2,VPAR(5)),(QT3,VPAR(6)),
     .            (AA,VPAR(7)),(BB,VPAR(8)),(CC,VPAR(9)),
     .            (HEATN,VPAR(10)),(HEATM,VPAR(11)),(HEATA,VPAR(12)),
     .            (PHCL,VPAR(13)),(PCO,VPAR(14)),(PCO2,VPAR(15)),
     .            (PAL2O3,VPAR(16)),(PNO,VPAR(17)),(GAMMAX,VPAR(18))
      DIMENSION ILINE(32),IDATAF(10),IERS(32),IMAPL(48),NAMEF(3)
C
C       INPUT FORMAT STATEMENT
C
  100 FORMAT (I2,1XA1)
C
C       OUTPUT FORMAT STATEMENT
C
C
C       DIMENSION STATEMENT
C
      DIMENSION IX(8),IY(8)
C
C       DATA STATEMENTS
C
      DATA IERS/32*2H  /
      DATA NAMEF/2H?R,2HIS,2HO2/
      DATA IMAPL/2H40,2H,S,2HEA,2H M,2HAP,2H  ,
     1           2H40,2H,L,2HAN,2HD ,2HMA,2HP ,
     1           2H41,2H,S,2HEA,2H M,2HAP,2H  ,
     1           2H41,2H,L,2HAN,2HD ,2HMA,2HP ,
     1           2H17,2H,S,2HEA,2H M,2HAP,2H  ,
     1           2H17,2H,L,2HAN,2HD ,2HMA,2HP ,
     1           2H39,2H,S,2HEA,2H M,2HAP,2H  ,
     1           2H39,2H,L,2HAN,2HD ,2HMA,2HP /
      DATA LCHAR/1HL/
      DATA IX/5450,5411,4030,4025,8750,8730,4100,4100/
      DATA IY/2630,8243,2465,8050,2990,8600,1700,7300/
C
C       IS THIS THE FIRST TIME THROUGH THIS SUBROUTINE? --
C       IF NOT, IT IS NOT NECESSARY TO CALCULATE THE INDEX OF THE
C       COORDINATES, I, AGAIN
C
      IF(IGOTMP .NE. 0)GO TO 7
C
C       THIS IS THE FIRST TIME THROUGH -- READ IN THE COMPLEX NUMBER
C       AND THE DESIRED MAP, i.e. SEA OR LAND
C
      CALL DREAD(NAMEF,7,ILINE)
```

```
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(0.,YPOS,0,ILINE,64,0,0)
      YPOS = YPOS - 16.
      IF(LVERSN.EQ.0)GO TO 10
      IF(YPOS.LT.48.) YPOS = 458.
   10 IF(IVHICL.EQ.1) CALL DREAD(NAMEF,8,ILINE)
      IF(IVHICL.EQ.2) CALL DREAD(NAMEF,9,ILINE)
      IF(IVHICL.EQ.0) CALL DREAD(NAMEF,10,ILINE)
      CALL LERS(YPOS,LVERSN)
      CALL CHAR(24.,YPOS,0,ILINE(2),8,3,0)
      CALL CHAR(95.,YPOS,0,ILINE(7),50,0,0)
      CALL IN(1,JTYPE,0.,0.,0,0,0,0,31,0,31,IXC,IYC)
      CALL CHAR(0.,YPOS,0,IERS,64,0,0)
      CALL CHAR(200.,YPOS+16,0,IERS,25,0,0)
      IF(IXC.LT.6.AND.IVHICL.EQ.1) I=1
      IF(IXC.GT.5.AND.IXC.LT.12.AND.IVHICL.EQ.1) I=2
      IF(IXC.GT.11.AND.IXC.LT.18.AND.IVHICL.EQ.1) I=3
      IF(IXC.GT.17.AND.IVHICL.EQ.1) I=4
      IF(IXC.LT.6.AND.IVHICL.EQ.2) I=5
      IF(IXC.GE.6.AND.IVHICL.EQ.2) I=6
      IF(IXC.LT.6.AND.IVHICL.EQ.0) I=7
      IF(IXC.GE.6.AND.IVHICL.EQ.0) I=8
      IMP = (I - 1)*6 + 1
      CALL CHAR(208.,YPOS+16.,0,IMAPL(IMP),12,0,0)
      YPOS = YPOS - 16.
      IF(LVERSN.EQ.0)GO TO 6
      IF(YPOS .LT.48.) YPOS = 458.
C
C        SET THE COORDINATES BASED ON THE INDEX I
C
C*** SET PAD INDEX FOR FUTURE USE...
    6 IGOTMP = I
7     CONTINUE
C
C*** RELOAD PAD INDEX PREVIOUSLY COMPUTED
      I = IGOTMP
      IX0 = IX(I)
      IY0 = IY(I)
C
C        RETURN TO THE CALLING PROGRAM
C
      RETURN
C
C        END OF ORGIN
C
      END
      SUBROUTINE SYMBL(IWIDE,IHI,ISYMB,LUPLT)
C
C   -----------------------------------------------------------------
C   -                                                               -
C   -  THIS SUBROUTINE PLOTS A GIVEN SYMBOL A SPECIFIED WIDTH AND   -
C   -  HEIGHT.                                                      -
C   -                                                               -
C   -----------------------------------------------------------------
C
      IX=-IWIDE/2
      IY=-IHI/2
      WRITE(LUPLT) -1,-1,IX,IY
      WRITE(LUPLT,100) IWIDE,0,0,IHI,ISYMB
  )      MAT(    .A1,
```

```
      IY=-IY
      WRITE(LUPLT)-1,-1,IX,IY
      RETURN
      END
      SUBROUTINE DREAD(NAMEF,LNUM,ILINE)
C
C  ---------------------------------------------------------------------
C  -                                                                   -
C  -  THIS SUBROUTINE READS A SPECIFIED QUESTION FROM A SPECIFIED      -
C  -  QUESTION FILE. THE QUESTION IS THEN DISPLAYED UPON THE           -
C  -  PLASMASCOPE FOR PROCESSING.                                      -
C  -                                                                   -
C  ---------------------------------------------------------------------
C
      DIMENSION NAMEF(3),IDCB(144),IBUF(40),ILINE(32)
      CALL OPEN(IDCB,IERR,NAMEF,0)
      LOOP = LNUM - 1
      DO 10 I=1,LOOP
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
10    CONTINUE
      CALL BLANK(IBUF,40)
      CALL READF(IDCB,IERR,IBUF)
      CALL CODE
      READ(IBUF,100) (ILINE(I),I=1,32)
100   FORMAT(32A2)
      CALL CLOSE(IDCB,IERR)
      RETURN
      END
      SUBROUTINE BLANK(IBUF,II)
C
C  ---------------------------------------------------------------------
C  -                                                                   -
C  -  THIS SUBROUTINE FILLS A SPECIFIED ARRAY WITH A SPECIFIED         -
C  -  NUMBER OF BLANKS.                                                -
C  -                                                                   -
C  ---------------------------------------------------------------------
C
      DIMENSION IBUF(40)
      DATA IBLK/2H  /
      DO 10 I=1,II
10    IBUF(I) = IBLK
      RETURN
      END
      SUBROUTINE LERS(YPOS,LVERSN)
C
C  ---------------------------------------------------------------------
C  -                                                                   -
C  -  THIS SUBROUTINE ERASES A SPECIFIED LINE OF 64-CHARACTERS         -
C  -  FROM THE PLASMASCOPE.                                            -
C  -                                                                   -
C  ---------------------------------------------------------------------
C
      DIMENSION IERS(32)
      DATA IERS/32*2H  /
      IF(YPOS .GT. 48.0)GO TO 4
      IF(LVERSN.EQ.0)GO TO 4
      YPOS = 458.0
      CALL CLEFF
      RETURN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
4 CALL CHAR(0.0,YPOS,0,IERS,64,0,0)
  CALL CHAR(0.0,YPOS-16.0,0,IERS,64,0,0)
  RETURN
  END
  END$
```

4.2 AUXILIARY SUBROUTINE LISTINGS

This section contains the current REEDS auxiliary subroutine listings. They are listed according to the sequence indicated in Section 4.0.

```
&CTLB2 T=00003 IS ON CR00003 USING 00015 BLKS R=0000

FTN4,L
      SUBROUTINE GRAF(ILOCK)    ,    REEDS   CRT   GRAPHICS   *CAB,810313
      RETURN
      END
      SUBROUTINE NGRAF
      RETURN
      END
      SUBROUTINE CLEAR
      COMMON IDUM(64),LU,IDUM1(492)
C  ξY
  200 FORMAT ("ξE")
C  ξZ
      WRITE (LU,200)
      CALL DELAY(0.1)
      RETURN
      END
      SUBROUTINE CHAR(X,Y,IXY,ICHAR,NCHAR,MODE,IALT)
      COMMON IDUM(64),LU,IDUM1(492)
C  ξY
  200 FORMAT ("ξ&a"I2"c"I2"Rξ&d@"32A2)
  201 FORMAT ("ξ&a"I2"c"I2"Rξ&dB"32A2"ξ&d@")
C  ξZ
      INTEGER BLANKS,CHOUT(32)
      DIMENSION ICHAR(1)
      DATA BLANKS/2H  /
      IF(IXY .GE. 0)GO TO 2
      IX = INT(X)
      IY = 31 - INT(Y)
      GO TO 7
    2 IX = INT(0.125 * X)
      IY = 31 - INT(0.0625 * Y)
    7 NWORDS = (NCHAR - 1)/2 + 1
      IF(MODE .NE. 1)GO TO 15
      DO 12 I=1,NWORDS
   12 CHOUT(I) = BLANKS
      GO TO 19
   15 DO 16 I=1,NWORDS
   16 CHOUT(I) = ICHAR(I)
   19 IF(MODE .EQ. 3)GO TO 25
      WRITE (LU,200) IX,IY,(CHOUT(I),I=1,NWORDS)
      GO TO 28
   25 WRITE (LU,201) IX,IY,(CHOUT(I),I=1,NWORDS)
   28 CALL DELAY(0.1)
      RETURN
      END
      SUBROUTINE DELAY(SEC)
      MSEC = 1000.0 * SEC
      CALL WAIT(MSEC,1)
      RETURN
      END
      SUBROUTINE WAIT(MULT,IRES,IERR)
C
C        THIS ROUTINE DUPLICATES THE FUNCTION OF THE WAIT SUBROUTINE
C        FOUND IN THE ISA LIBRARY.  THE INPUT VARIABLES ARE DEFINED
C        AS FOLLOWS:
C             MULT - POSITIVE INTEGER INDICATING THE NUMBER OF UNITS
C                    THE CALLING PROGRAM SHOULD BE PUT IN  WAIT STATE
C             IRES - THE RESOLUTION OF MULT, I.E.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C                            0 - 10'S OF MILLISECONDS
C                            1 - MILLISECONDS
C                            2 - SECONDS
C                            3 - MINUTES
C                 IERR - RETURNED ERROR FLAG
C                            1 - REQUEST ACCEPTED
C                            3 - ILLEGAL PARAMETER
C
      IF(MULT.GT.0 .AND. IRES.GE.0 .AND. IRES.LE.3)GO TO 4
      IERR = 3
      RETURN
    4 IERR = 1
      IM = MULT
      IF(IRES .EQ. 1)IM = IM/10
      IR = IRES
      IF(IRES .EQ. 0)IR = 1
      CALL EXEC(12,0,IR,0,-IM)
      RETURN
      END
      SUBROUTINE IN(ITYPE,JTYPE,X,Y,IXY,ICHAR,NCHAR,IXL,IXH,IYL,IYH,
     .                IX,IY)
      COMMON IDUM(64),LU,IDUM1(492)
  100 FORMAT (32A2)
  101 FORMAT (I2)
      INTEGER UP0,UP9,UPBL
      DIMENSION ICHAR(1)
      DATA UP0/030000B/, UP9/034400B/, UPBL/020000B/,
     .     LOW0/000060B/, LOW9/000071B/, LOWBL/000040B/
      JTYPE = 0
      IF(ITYPE .EQ. 1)GO TO 4
      NWORDS = (NCHAR - 1)/2 + 1
      READ (LU,100) (ICHAR(I),I=1,NWORDS)
      IF(ITYPE .EQ. 0)RETURN
      IXIN = ICHAR(1)
      GO TO 7
    4 READ (LU,100) IXIN
    7 IXUP = IAND(IXIN,177400B)
      IF(IXUP .EQ. UPBL)IXUP = UP0
      IXLOW = IAND(IXIN,000377B)
      IF(IXLOW .NE. LOWBL)GO TO 9
      IXLOW = ISHIF(IXUP,8)
      IXUP = UP0
    9 IF(IXUP.GE.UP0 .AND. IXUP.LE.UP9 .AND.
     .   IXLOW.GE.LOW0 .AND. IXLOW.LE.LOW9)GO TO 12
      IF(ITYPE .EQ. 2)RETURN
      GO TO 4
   12 IXIN = IOR(IXUP,IXLOW)
      CALL CODE
      READ (IXIN,101) IX
      IF(IX.GE.IXL .AND. IX.LE.IXH)GO TO 15
      IF(ITYPE .EQ. 2)RETURN
      GO TO 4
   15 JTYPE = 1
      RETURN
      END
      SUBROUTINE VERSN(IVER)
      IVER = 1
      RETURN
      END
```

```
&MLIB2 T=00003 IS ON CR00003 USING 00018 BLKS R=0000

FTN4,L
      SUBROUTINE CHAR(X,Y,IXY,ICHAR,NCHAR,MODE,ISIZE)
     .                 ,  REEDS    PLOTTER GRAPHICS          *CAB,810815
      COMMON IDUM(64),LU,IDUM1(492)
      DIMENSION ICHAR(32),JCHAR(33)
      IF(MODE .EQ. 1)RETURN
      JCHAR(1) = NCHAR
      DO 5 K=2,33
5     JCHAR(K) = ICHAR(K-1)
      XX= X/51.2
      YY= Y/51.2
      SIZE = 10./64.
      IF(ISIZE.EQ.1) SIZE = SIZE/1.5
      CALL SYMB(XX,YY,SIZE,JCHAR,0.,1)
      RETURN
      END
      SUBROUTINE LINE(X,Y,NXY,MODE)
      DIMENSION X(1),Y(1)
      IF(NXY.LT.2 .OR. MODE.NE.0)RETURN
      XX = X(1)/51.2
      YY = Y(1)/51.2
      CALL PLOT(XX,YY,3)
      DO 7 I=2,NXY
      XX = X(I)/51.2
      YY = Y(I)/51.2
    7 CALL PLOT(XX,YY,2)
      RETURN
      END
      SUBROUTINE GRAF(IL)
      CALL PLTLU(12)
      CALL LLEFT
      RETURN
      END
      SUBROUTINE NGRAF
      CALL URITE
      RETURN
      END
      SUBROUTINE CLEAR
     .COMMON IDUM(64),LU,IDUM1(492)
C  {Y
  200 FORMAT ("{E")
C  {Z
      WRITE (LU,200)
      CALL DELAY(0.1)
      RETURN
      END
      SUBROUTINE LALT(I1,I2,I3)
      RETURN
      END
      SUBROUTINE CHARC(X,Y,IXY,ICHAR,NCHAR,MODE,IALT)
      COMMON IDUM(64),LU,IDUM1(492)
C  {Y
  200 FORMAT ("{&a"I4"c"I4"R"32A2)
  201 FORMAT ("{&a"I4"c"I4"R{&dB"32A2)
C  {Z
      INTEGER BLANKS,CHOUT(32)
      DIMENSION ICHAR(1)
        fn b    S/21
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF(IXY .GE. 0)GO TO 2
      IX = INT(X)
      IY = 31 - INT(Y)
      GO TO 4
    2 IX = INT(0.125 * X)
      IY = 31 - INT(0.0625 * Y)
    4 IF(IY .LE. 22)GO TO 7
      IY = IY - 22
      GO TO 4
    7 NWORDS = (NCHAR - 1)/2 + 1
      IF(MODE .NE. 1)GO TO 15
      DO 12 I=1,NWORDS
   12 CHOUT(I) = BLANKS
      GO TO 19
   15 DO 16 I=1,NWORDS
   16 CHOUT(I) = ICHAR(I)
   19 IF(MODE .EQ. 3)GO TO 25
      WRITE (LU,200) IX,IY,(CHOUT(I),I=1,NWORDS)
      GO TO 28
   25 WRITE (LU,201) IX,IY,(CHOUT(I),I=1,NWORDS)
   28 CALL DELAY(0.1)
      RETURN
      END
      SUBROUTINE DELAY(SEC)
      MSEC = 1000.0 * SEC
      CALL WAIT(MSEC,1)
      RETURN
      END
      SUBROUTINE WAIT(MULT,IRES,IERR)
C
C        THIS ROUTINE DUPLICATES THE FUNCTION OF THE WAIT SUBROUTINE
C        FOUND IN THE ISA LIBRARY.  THE INPUT VARIABLES ARE DEFINED
C        AS FOLLOWS:
C              MULT - POSITIVE INTEGER INDICATING THE NUMBER OF UNITS
C                        THE CALLING PROGRAM SHOULD BE PUT IN A WAIT STATE
C              IRES - THE RESOLUTION OF MULT, I.E.
C                               0 - 10'S OF MILLISECONDS
C                               1 - MILLISECONDS
C                               2 - SECONDS
C                               3 - MINUTES
C              IERR - RETURNED ERROR FLAG
C                               1 - REQUEST ACCEPTED
C                               3 - ILLEGAL PARAMETER
C
      IF(MULT.GT.0 .AND. IRES.GE.0 .AND. IRES.LE.3)GO TO 4
      IERR = 3
      RETURN
    4 IERR = 1
      IM = MULT
      IF(IRES .EQ. 1)IM = IM/10
      IR = IRES
      IF(IRES .EQ. 0)IR = 1
      CALL EXEC(12,0,IR,0,-IM)
      RETURN
      END
      SUBROUTINE IN(ITYPE,JTYPE,X,Y,IXY,ICHAR,NCHAR,IXL,IXH,IYL,IYH,
     .                IX,IY)
      COMMON IDUM(64),LU,IDUM(232)
  100 FORMAT (32A2)
  101 FORMAT (I2)
```

```
      INTEGER UP0,UP9,UPBL
      DIMENSION ICHAR(1)
      DATA UP0/030000B/, UP9/034400B/, UPBL/020000B/,
     .     LOW0/000060B/, LOW9/000071B/, LOWBL/000040B/
      JTYPE = 0
      IF(ITYPE .EQ. 1)GO TO 4
      NWORDS = (NCHAR - 1)/2 + 1
      READ (LU,100) (ICHAR(I),I=1,NWORDS)
      IF(ITYPE .EQ. 0)RETURN
      IXIN = ICHAR(1)
      GO TO 7
    4 READ (LU,100) IXIN
    7 IXUP = IAND(IXIN,177400B)
      IF(IXUP .EQ. UPBL)IXUP = UP0
      IXLOW = IAND(IXIN,000377B)
      IF(IXLOW .NE. LOWBL)GO TO 9
      IXLOW = ISHIF(IXUP,8)
      IXUP = UP0
    9 IF(IXUP.GE.UP0 .AND. IXUP.LE.UP9 .AND.
     .   IXLOW.GE.LOW0 .AND. IXLOW.LE.LOW9)GO TO 12
      IF(ITYPE .EQ. 2)RETURN
      GO TO 4
   12 IXIN = IOR(IXUP,IXLOW)
      CALL CODE
      READ (IXIN,101) IX
      IF(IX.GE.IXL .AND. IX.LE.IXH)GO TO 15
      IF(ITYPE .EQ. 2)RETURN
      GO TO 4
   15 JTYPE = 1
      RETURN
      END
      SUBROUTINE VERSN(IVER)
      IVER = 1
      RETURN
      END
      END$
```

```
*CALIX T=00003 IS ON CR00037 USING 00013 BLKS R=0000

ASMB,R,F,L
*
*      ENTRY CALIX CONVERTS AN X COORDINATE IN USER'S UNITS
*      (IN A & B REGISTERS) TO AN X COORDINATE IN RASTER UNITS
*      (IN A REGISTER) FOR THE PLASMASCOPE
*
*      ENTRY CALIY CONVERTS A Y COORDINATE IN USER'S UNITS
*      (IN A & B REGISTERS) TO A Y COORDINATE IN RASTER UNITS
*      (IN A REGISTER) FOR THE PLASMASCOPE
*
*      ENTRY SETOR SETS THE X AND . ORIGIN FOR CONVERSION OF
*      USER'S UNITS TO RASTER UNITS
*
*      ENTRY SETSC SETS THE X AND Y SCALE FACTORS FOR CONVERSION
*      OF USER'S UNITS TO RASTER UNITS
*
*      ENTRY GETOR RETURNS THE X AND Y ORIGIN TO THE USER
*      AND SETS THE X AND Y ORIGIN BOTH TO 0
*
*      ENTRY GETSC RETURNS THE X AND Y SCALE FACTORS TO THE USER
*      AND SET THE X AND Y SCALE FACTORS BOTH TO 1
*
*      ENTRY LASTM IS THE STORAGE LOCATION CONTAINING THE
*      LAST MODE WORD OUTPUT
*
       NAM CALIX,7    REEDS    GRAPHICS                  *CAB,010811
       ENT CALIX,CALIY,SETOR,SETSC,GETOR,GETSC,LASTM
       EXT .ENTR
*
XTMP   BSS 1
YTMP   BSS 1
*
CALIX  NOP            ENTRY EXIT LINE
*
       FSB XORG       IX = (X - XORG) * XSCAL + 0.5
       FMP XSCAL
       FAD PT5
       FIX
       JMP CALIX,I    RETURN TO CALLING PROGRAM
*
*
*
*
CALIY  NOP            ENTRY/EXIT LINE
*
       FSB YORG       IY = (Y - YORG) * YSCAL + 0.5
       FMP YSCAL
       FAD PT5
       FIX
       JMP CALIY,I    RETURN TO CALLING PROGRAM
*
*
*
*
SETOR  NOP            ENTRY/EXIT LINE
*
       JSB .ENTR      GET ADDRESSES OF XORG,YORG
       DEF XTMP
```

```
*
       DLD XTMP,I    STORE XORG
       DST XORG
*
       DLD YTMP,I    STORE YORG
       DST YORG
*
       JMP SETOR,I   RETURN TO CALLING PROGRAM
*
*
*
*
SETSC  NOP           ENTRY/EXIT LINE
*
       JSB .ENTR     GET ADDRESSES OF XSCAL,YSCAL
       DEF XTMP
*
       DLD XTMP,I    STORE XSCAL
       DST XSCAL
*
       DLD YTMP,I    STORE YSCAL
       DST YSCAL
*
       JMP SETSC,I   RETURN TO CALLING PROGRAM
*
*
*
*
GETOR  NOP           ENTRY/EXIT LINE
*
       JSB .ENTR     GET ADDRESSES OF XORG, YORG
       DEF XTMP
*
       DLD XORG      RETURN XORG AND REPLACE WITH 0
       DST XTMP,I
       DLD FLT0
       DST XORG
*
       DLD YORG      RETURN YORG AND REPLACE WITH 0
       DST YTMP,I
       DLD FLT0
       DST YORG
*
       JMP GETOR,I   RETURN TO CALLING PROGRAM
*
*
*
*
GETSC  NOP           ENTPY EXIT LINE
*
       JSB .ENTR     GET ADDRESSES OF XSCAL, YSCAL
       DEF XTMP
*
       DLD XSCAL     RETURN XSCAL AND REPLACE WITH 1
       DST XTMP,I
       DLD FLT1
       DST XSCAL
*
       DLD YSCAL     RETURN YSCAL AND REPLACE WITH 1
       . . YT. ,.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      DLD FLT1
      DST YSCAL
*
      JMP GETSC,I   RETURN TO CALLING PROGRAM
*
*
*
*
LASTM NOP           LAST MODE WORD OUTPUT
*
*
*
*
PT5   DEC 0.5
FLT0  DEC 0.0
FLT1  DEC 1.0
*
XORG  DEC 0.0
YORG  DEC 0.0
*
XSCAL DEC 1.0
YSCAL DEC 1.0
*
*
*
*
      END CALIX
```

```
&DLINE T=00003 IS ON CR00003 USING 00025 BLKS R=0000

ASMB,R,F,L
*
*      THIS ROUTINE PLOTS DASHED LINES ON THE PLASNASCOPE
*      WITH THE LENGTH OF THE DASHES AND THE LENGTH OF THE
*      SPACES VARIABLE
*
       NAM DLINE,7     REEDS     GRAPHICS                        *CAB,810310
       ENT DLINE
       EXT .ENTR,GETOR,GETSC,SQRT,LINE,SETOR,SETSC
*
X      BSS 1
Y      BSS 1
NXY    BSS 1
MODE   BSS 1
LDASH  BSS 1
LSPC   BSS 1
*
DLINE  NOP             ENTRY/EXIT LINE
*
       JSB .ENTR       GET ADDRESSES OF X,Y,NXY,ETC.
       DEF X
*
       LDA NXY,I       IS NXY < 2 ?
       ADA NEG1
       CMA,INA
       SSA,RSS
       JMP DLINE,I     YES -- RETURN TO CALLING PROGRAM
       STA NUM         NO -- SET POINT COUNTER
*
       JSB GETOR       GET ORIGIN VALUES AND SET THE ORIGIN
       DEF *+3          VALUES EQUAL TO 0 FOR CALLS TO THE
       DEF XORG         EXTERNAL ROUTINES CALIX AND CALIY
       DEF YORG
*
       JSB GETSC       GET SCALE VALUES AND SET THE SCALE
       DEF *+3          VALUES EQUAL TO 1 FOR CALLS TO THE
       DEF XSCAL        EXTERNAL ROUTINES CALIX AND CALIY
       DEF YSCAL
*
       LDA LDASH,I     STORE DASH LENGTH AND SPACE
       STA LENS         LENGTH IN LENS ARRAY
       LDA LSPC,I
       STA LENS+1
*
       CLA             CLEAR LEFTOVER LENGTH AND THE
       STA LOLEN        DASH OR SPACE INDICATOR
       STA DORS
*
       LDA MODE,I      STORE THE MODE
       STA MODED
*
       LDA X,I         GET, CONVERT TO RASTER UNITS,
       ISZ X            AND STORE THE FIRST POINT ON
       LDB X,I          THE LINE AS POINT 1
       ISZ X
       JSB CALIX
           XX
```

```
        LDA Y,I
        ISZ Y
        LDB Y,I
        ISZ Y
        JSB CALIY
        DST YY
*
NEXT    LDA X,I         GET, CONVERT TO RASTER UNITS,
        ISZ X            AND STORE THE NEXT POINT ON
        LDB X,I          THE LINE AS POINT 2
        ISZ X
        JSB CALIX
        DST XNEXT
*
        LDA Y,I
        ISZ Y
        LDB Y,I
        ISZ Y
        JSB CALIY
        DST YNEXT
*
SAME    FSB YY          DETERMINE THE LENGTH FROM
        FMP A            POINT 1 TO POINT 2
        DST TEMP
        DLD XNEXT
        FSB XX
        FMP A
        FAD TEMP
        JSB SQRT
        NOP
        FAD PT5         INTEGERIZE AND STORE IN ALEN
        FIX
        STA ALEN
        FLT             FLOAT AND STORE IN TEMP
        DST TEMP
*
        LDA ALEN        ADD IN ANY LEFTOVER LENGTH FROM
        ADA LOLEN        THE LAST TIME AND STORE IN LOLEN
        STA LOLEN
*
        CMA,INA         COMPARE WITH DESIRED LENGTH OF
        LDY DORS         DASH OR SPACE
        LBY LENS
        ADA B
        SSA
        JMP TOFAR       POINTS TOO FAR APART
        STA TEMP
*
        DLD XNEXT       POINTS TOO CLOSE TOGETHER OR EXACT
        DST XX+2
        DLD YNEXT
        DST YY+2
*
        JSB LINE        OUTPUT POINT 1 AND POINT 2
        DEF *+5
        DEF XX
        DEF YY
        DEF TWO
        DEF MODED
*
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      LDA TEMP       IF POINTS EXACT -- CHANGE INDICATORS
      SZA,RSS
      JSB CHANG
*
      DLD XX+2       MOVE POINT 2 INTO POINT 1
      DST XX
      DLD YY+2
      DST YY
*
      ISZ NUM        ANY MORE POINTS?
      JMP NEXT       YES -- GET ANOTHER
      JSB SETOR      NO -- RESET THE ORIGIN FOR THE EXTERNAL
      DEF *+3         CALLS TO CALIX AND CALIY
      DEF XORG
      DEF YORG
*
      JSB SETSC      RESET THE SCALE FOR THE EXTERNAL
      DEF *+3         CALLS TO CALIX AND CALIY
      DEF XSCAL
      DEF YSCAL
*
      JMP DLINE,I    RETURN TO CALLING PROGRAM
*
TOFAR ADA ALEN       POINTS TOO FAR APART --
      FLT             CALCULATE AN INTERMEDIATE POINT
      FDV TEMP        AND STORE IT
      DST TEMP
*
      DLD XNEXT
      FSB XX
      FMP TEMP
      FAD XX
      FAD PT5
      FIX
      FLT
      DST XX+2
*
      DLD YNEXT
      FSB YY
      FMP TEMP
      FAD YY
      FAD PT5
      FIX
      FLT
      DST YY+2
*
      JSB LINE       OUTPUT POINT 1 AND THE
      DEF *+5         INTERMEDIATE POINT
      DEF XX
      DEF YY
      DEF TWO
      DEF MODED
*
      JSB CHANG      CHANGE INDICATORS
*
      DLD XX+2       MOVE INTERMEDIATE POINT INTO
      DST XX          POINT 1 -- POINT 2 REMAINS THE
      DLD YY+2        SAME
      DST YY
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
       DLD YNEXT     TRY AGAIN TO GET TO POINT 2
       JMP SAME
*
*
*
*
CHANG NOP           SUBPROGRAM TO CHANGE INDICATORS
*
      LDA MODED     SWITCH THE MODE FROM WRITE TO
      CMA,INA        ERASE OR VICE-VERSA
      INA
      STA MODED
*
      LDA DORS      SWITCH THE DASH OR SPACE INDICATOR
      CMA,INA
      INA
      STA DORS
*
      CLA           CLEAR THE LEFTOVER LENGTH
      STA LOLEN
*
      JMP CHANG,I   RETURN TO THE APPROPRIATE PLACE
*
*
*
*
CALIX NOP           SUBPROGRAM TO CONVERT AN X
      FSB XORG       COORDINATE TO RASTER UNITS BY
      FMP XSCAL      X = AINT((X - XORG) * XSCAL + 0.5)
      FAD PT5        (SIMILAR TO EXTERNAL CALIX USED
      FIX            BY ALL OTHER ROUTINES)
      FLT
      JMP CALIX,I
*
*
*
*
CALIY NOP           SUBPROGRAM TO CONVERT A Y
      FSB YORG       COORDINATE TO RASTER UNITS BY
      FMP YSCAL      Y = AINT((Y - YORG) * YSCAL + 0.5)
      FAD PT5        (SIMILAR TO EXTERNAL CALIY USED
      FIX            BY ALL OTHER ROUTINES)
      FLT
      JMP CALIY,I
*
*
*
*
A     EQU 0
B     EQU 1
*
NEG1  DEC -1
TWO   DEC 2
PT5   DEC 0.5
*
XORG  BSS 2
YORG  BSS 2
*
XSCAL BSS 2
YSCAL BSS 2
```

```
*
NUM   NOP
LOLEN NOP
DOES  NOP
MODED NOP
ALEN  NOP
*
LENS  BSS 2
*
XX    BSS 4
YY    BSS 4
*
XNEXT BSS 2
YNEXT BSS 2
*
TEMP  BSS 2
*
      END DLINE
```

ORIGINAL PAGE IS
OF POOR QUALITY

4.3 MAIN MODULE QUESTION FILES

This section contains the Listings of the Main Module Question Files. They are listed according to the sequence indicated in Section 4.0

```
?REEDS T=00003 IS ON CR00031 USING 00006 BLKS R=0000

1234567890123456789012345678901234567890123456789012345678901234
  *****NASA/MSFC EXPERIMENTAL MULTILAYER DIFFUSION MODEL*****
  USE DEFAULT COMMON DATA FILE (DATA01) ?            YES       NO
  Enter Number of Runs and Common Data File (e.g. 04,DATA02)
       Keyboard Entry:
  Run    Will Use Data File
  Run Type:       RESEARCH=R    OPERATIONAL     PRODUCTION=P
  Default Launch Time/Date?                           YES       NO
  Enter Launch Time/Date:
  Select Launch Vehicle:   DELTA=D       TITAN=T     SHUTTLE=S
  Enter Top of Surface Layer(M):
  Enter Sigma of Wind Azimuth Angle (xx.x):
  ***** PROGRAM REEDS HAS TERMINATED NORMALLY *****
  Desired Processing ?         TERMINATE=T     RESTART=R
  Do You Want Concentrations At:
      SURFACE=SU    STABILIZATION=ST    SOMETHING ELSE=SE
  Use Default Height of Calculation? Zz=          YES=25    NO=30
       Enter Zz:
  Use Default Surface Reflection ?   Rf=1.0      YES=25    NO=30
  Rf= .8       .7       .5       .3       .1        .0    Enter:
  Use Default Height of Base Layer ? Zb=0.0      YES=25    NO=30
       Enter Zb:
  MET Plots?                                          YES       NO
  Generate The MET Plot Form?                         YES       NO
  Generate Printer Output?                            YES       NO
  Enter Specific Sounding: (i.e. 06,JAN, 515Z)
      Sounding #1:
      Sounding #2:
      Sounding #3:
      Sounding #4:
      Sounding #5:
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
?RCLDS T=00003 IS ON CR00003 USING 00002 BLKS R=0000

1234567890123456789012345678901234567890123456789012345678901234
  Estimated Top of Surface Layer(M) is:
  Use:  EST=05      MET PROFILE TO COMPUTE=10 ENTER VALUE:
  Enter Desired Top of Surface Layer(M):
  Calculated Top of Surface Transport Layer is:
  Use a Sigma for Wind Azimuth Angle of:       ?  YES      NO
  Enter Desired Sigma of Wind Azimuth Angle  (xx.x):
```

Listing 2B

```
?RCONS T=00003 IS ON CR00003 USING 00003 BLKS R=0000

1234567890123456789012345678901234567890123456789012345678901234
  Centerline Concentration Plot Desired ?         YES      NO
  Specific Site Predictions (Plotted)?            YES      NO
  Concentration Isopleth Plot Desired?            YES      NO
  Enter Range(m) and Azimuth(deg): (0 Terminates Procedure)
       Range:               Azimuth:
  Enter Desired Complex:   (Sea or Land Map)
    40,S=05     40,L=08      41,S=15     41,L=18
    17,S=05     17,L=08
    39,S=05     39,L=08
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
?RISOS T=00003 IS ON CR00003 USING 00002 BLKS R=0000

0001  1234567890123456789012345678901234567890123456789012345678901234
0002    Default Isopleth Concentration Values are:  Conc(1)=
0003        Conc(2)=            Conc(3)=            Conc(4)= -1.0
0004    Use Default Isopleth Concentration Values ?      YES        NO
0005    Enter Isopleth Values: (Negative Value Terminates Procedure)
0006    Enter: Conc( )=
0007    Enter Desired Complex:  (Sea or Land Map)
0008      40,S=05      40,L=08      41,S=15      41,L=20
0009      17,S=05      17,L=08
0010      39,S=05      39,L=08
```

4.4 EXAMPLE METEOROLOGICAL SOUNDING FILES

This section contains the current listings of two (2) example meteorological sounding files. They are listed according to the sequence indicated in Section 4.0.

```
DATA01 T=00003 IS ON CR00031 USING 00007 BLKS R=0000

TEST NBR 03717 04834 0-24HR
RAWINSONDE RUN AN/GMD-4
CAPE CANAVERAL AFS, FLORIDA
1200Z  16 NOV 1969
ASCENT NBR   0543
AAALT   DIR SPD  TEMP  D/PT  PRESS   RH ABHUM DENSITY  IR V/S   SHEAR
AAAFT   DEG KTS DEG C DEG C   MBS   PCT  G/M3   G/M3    N KTS /SEC DEG
000016 320 000  07.1   23.2 1027.00 069 20.34 1273.00 377 678 0000 000
000735  10 016  09.6   22.9 1000.00 082 20.23 1136.98 372 674 .007 104
000754 010 016  09.7   21.7 0999.30 080 18.98 1107.97 359 671 .003 201
002126 027 014  06.5   16.6 0950.00 070 13.09 1078.02 323 670 .002 203
002172 028 014  06.4   15.3 0940.50 072 12.84 1048.16 310 668 .003 308
003579 007 008  06.3   13.6 0900.00 071 11.58 1017.62 296 666 .003 307
003629 006 006  06.3   13.6 0898.60 084 11.70 0990.98 292 663 .001 340
005121 316 006  07.2   10.8 0850.00 079 09.78 0963.41 274 661 .001 109
005960 335 004  07.7    8.7 0824.20 076 08.56 0935.27 261 659 .003 185
TERMINATION       26724 FT    8145 MTRS GEOP  374.0 MBS
TROPOPAUSE        0  FEET        0 MB       0 C      0 C    0 N
MANDATORY LEVELS
ALT FT DIR KTS  TEMP  D/PT PRESS    RH
006754 009 004   6.4  10.7 0800.0 078
008374 041 002   4.9   7.6 0753.5 080
008485 041 002   4.9   3.5 0750.0 071
012663 157 008   4.7  -2.2 0650.0 061
014789 152 010    .6  -6.9 0600.0 057
017062 155 009  -4.2 -11.9 0550.0 055
019510 179 008  -8.1 -27.5 0500.0 019
022171 122 001 -12.9 -16.7 0450.0 073
025087 036 007 -19.0  99.9 0400.0 999
SIGNIFICANT LEVELS
ALT FT DIR KTS  TEMP  D/PT PRESS    IR
011839 154 007   6.6    .2 0670.02 216
017247 156 008  -4.6 -12.1 0546.92 171
025525 039 007 -10.0 -46.3 0393.99 121
026796 999 999 -21.3 -45.9 0374.00 116
NNNN
```

ORIGINAL PAGE IS OF POOR QUALITY

```
DATA01 T=00004 IS ON CR00037 USING 00008 BLKS R=0036

0001  TEST NBR 03718 0 MINUS 40 MI
0002  RAWINSONDE RUN AN/GK -1
0003  CAPE CANAVERAL AFS, FLORIDA
0004  06312  10 DEC 1974
0005  ASCENT NBR   0759
0006    ALT  DIR SPD  TEMP  D/PT  PRESS   RH ABHUM DENSITY  IR V/S   SHEAR
0007     FT  DEG KTS DEG C DEG C   MBS   PCT  G/M3   G/M3    N KTS /SEC DEG
0008  000016 310 006  07.9  02.5 1023.00 069  5.64 1264.61 317 653 .000 000
0009  001000 345 021  08.9  03.3 0986.64 068  5.54 1215.12 308 654 .027 358
0010  002000 355 020  07.7  -0.2 0950.91 059  4.72 1176.88 292 653 .006 087
0011  003000 346 020  07.9  -4.7 0916.53 040  3.31 1134.21 273 653 .005 259
0012  004000 332 023  07.1  -5.7 0883.27 039  3.69 1096.03 263 652 .010 277
0013  005000 314 025  06.5  -7.1 0851.17 037  2.70 1058.64 253 651 .013 247
0014  006000 306 027  05.8  -7.0 0820.13 037  2.63 1022.52 244 651 .008 253
0015  007000 319 027  03.6  -8.6 0790.69 037  2.28 0933.15 236 648 .011 039
0016  008000 301 033  02.6 -10.5 0760.98 037  2.16 0960.23 228 647 .010 249
0017  009000 311 036  01.6 -11.1 0732.84 038  2.07 0927.90 220 646 .011 011
0018  010000 317 043  00.2 -11.9 0705.62 039  1.94 0897.96 212 644 .014 346
0019  TERMINATION       22682 FT    6914 MTRS GEOP  439.2 MBS
0020  TROPOPAUSE         0  FEET        0 MB       0 C       0 C     0 N
0021  MANDATORY LEVELS
0022  ALT FT DIR KTS  TEMP  D/PT PRESS    RH
0023  000633 339 019  09.1  03.1 1000.0  066
0024  002023 355 020  07.8  -0.5 0950.0  057
0025  003487 340 022  07.4  -5.2 0900.0  040
0026  005029 314 025  06.4  -7.2 0850.0  037
0027  006655 318 027  04.3  -9.1 0800.0  037
0028  008371 297 035  02.1 -10.5 0750.0  038
0029  SIGNIFICANT LEVELS
0030  ALT FT DIR KTS  TEMP  D/PT PRESS    IR
0031  000743 341 019  09.5  03.5  996.00 310
0032  001827 355 020  06.7  02.2 0957.00 299
0033  002169 354 020  08.6  -2.4 0945.00 284
0034  006136 305 027  05.8  -7.9 0916.00 243
0035  007343 310 030  03.0 -10.4 0780.00 233
0036  NNNN
```

ORIGINAL PAGE IS
OF POOR QUALITY